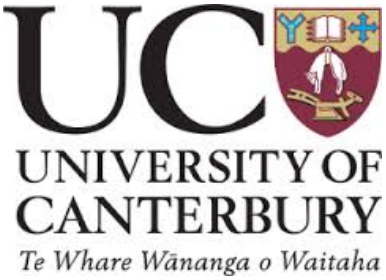
Statistical Methods for Correspondence Problems with Branched Structures

---

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Doctor of Philosophy
by
Davide Floriello



University of Canterbury
2015

Abstract

Correspondence problems are among the most studied topics in computer vision. They consist of two main steps: understanding the information present in images and associating features that represent the same information, between different images. Solving correspondence problems is a fundamental task in many applications, such as recovering a three dimensional structure from images. Correspondence problems become even more challenging when scenes show self-similar structures or have a high level of ambiguity. The aim of my research is to solve such problems in the context of self-similar branched structures, such as vine structures. The solution will be then implemented on an automated pruning robot.

Several methods to solve correspondence problems have already been proposed and they proved very effective with different scenes. However, when they are applied to vine images, because of the structure of the plants, their performances are reduced. I review and analyse some of those methods in Chapter 2.

A model-based approach proved effective in the present context. It enables decreasing the level of ambiguity of matches and it avoids making too many assumptions about the scene that could be violated with vine structures. First, in Chapter 3 I define and analyse variables describing branched structures.

Several algorithms are proposed and compared. Three state-of-the-art methods, RANSAC, graph matching and maximum likelihood, are suitably modified to be applied to my case. Those are then compared to the algorithm used in the initial state of the project and with three novel algorithms I proposed. These newly proposed methods compute the probabilities of a correspondence being correct given some variables of the branches. My main approach represents an improvement of 90% over prior research in terms of branches reconstructed, thus providing a more complete and accurate reconstructions. Moreover, it has better performance than state-of-the-art algorithms applied to this specific problem.

Table of Contents

List of Figures

List of Tables

Acknowledgements

There are many people that I would like to thank, for they helped me a lot, directly or indirectly, throughout all my time in New Zealand. First of all, my supervisors Richard Green and Tom Botterill, for their advices and feedback and for having given me the possibility of doing this PhD. I am especially thankful to Tom, because from him I learnt a lot in terms of technical knowledge and programming skills.

Secondly, I would like to thank the people that shared this PhD adventure with me, in the same office, who contributed to make the daily work definitely funnier.

Last but not least, all the friends and people met through the university, swimming and other activities, who added a lot of value to my time here.

Publications

Some aspects of the research presented here have previously appeared in refereed publications.

The first paper is a review on several approaches that were used to solve the correspondence problem in contexts similar to mine.

D. Floriello, T. Botterill, R. Green, Recovering and analyzing 3D models of branched structures using computer vision: a review. *Proceedings of IVCNZ 2012*, 325 - 329.

The second paper is about solving the correspondence problem defining a suitable geometric probability measure. The method was used on a simple artificial model of vine structures used at the beginning of my research.

D. Floriello, T. Botterill, R. Green, Defining a geometric probability measure in correspondence problems for branched structures. *Proceedings of IVCNZ 2013*, 311 - 316.

The third paper shows how including a constraint on the connectedness of the final structure improves on the correspondences selected on a test set of ground truth data.

D. Floriello, T. Botterill, R. Green, A topological constrained model-based approach to correspondence problems for branched structures. *Proceedings of IVCNZ 2014*, 60 - 66.

Chapter I

Introduction

1.1 General introduction

I propose new statistical methods to solve correspondence problems in the case of branched structures, exploiting knowledge of the structure itself. The main reasons for this approach are that I can overcome some of the limitations shown by previous methods if they are applied to self-similar branched structures, avoid further assumptions that may not be always true and design algorithms adapting to the particular structure of the problem.

Solving correspondence problems is an essential step in many applications, such as recovering the 3D reconstruction of an object, given a certain number of images of it. In the past, numerous methods have been already proposed as solutions. However, the applications of those methods or the technology used are not relevant to my application. This led to assumptions that do not hold in my case. I will further discuss this aspect in Chapter 2. Therefore, I cannot apply such methods directly to my problem: I need to modify them or to propose a new solution.

The practical application of my research is to be used on an automated robot whose aim is to prune vine plants. My methods for finding correspondences are used in the system for 3D reconstruction of the vines. The pruning system will use it to locate the different branches in space and finally cut them. Such an automated pruning system is being developed at the University of Canterbury and my research will therefore help its development, specifically, its vision apparatus. The automated pruning system being developed is a complex project: it involves many different areas of science and technology. The resulting machinery will be used to help the wine industry, a fast growing and now a key sector of New Zealand's economy.

1.2 Motivation

Mankind relies on sight more than any other sense. This fact permeates almost every aspect of everyday life, as people mainly use vision to communicate and for understanding purposes. A systematic study of visual information, namely images, was therefore initiated. In the 1970s, indeed, images became objects of scientific interest in their own right. Computer Vision is the part of applied mathematics that deals with images. Cao *et al.* state, "Computer Vision aims to give wherever possible a mathematical definition of visual perception" [25, p. 1]. The development of Computer Vision gave rise to new discoveries in many different fields, from Mathematics to Perception Theory and it also allowed scientists to design algorithms used in artificial systems. The versatility and accuracy of such machinery is extremely high, but there are still some fields where it bears no comparison to human perception. Years of evolution has been refining human vision in such a way that we can now reconstruct the spatiality of observed scenes even in case of depth discontinuities and occlusions. This is not true for artificial perception systems. In general, a human can recover the spatiality of a scene represented in an image, whereas a machine cannot do this by itself. It needs some other information and external criteria.

If a seeing machine can handle two or more different images of the same scene, a method for recovering three dimensionality is provided by Epipolar Geometry. I refer to Hartley and Zisserman [67] for a detailed explanation of the theory and application of Epipolar Geometry. When the coordinates of the same object in the different images are known, Epipolar Geometry allows a reconstruction of the spatial position of the pictured object. The problem with this approach is exactly in the way the corresponding features, concerning the same pictured object, are found. If the represented scene is simple, the identification of the same object in the different images can be easy. Then, the respective coordinates are known and the three dimensional reconstruction is straightforward. In the simple picture in Figure 1.1, it is clear how to recognize that the two sets of parallelepipeds are the same. For example it can be observed how the shading of the surfaces match, between the two images, or how the relative position of the two solids is preserved.

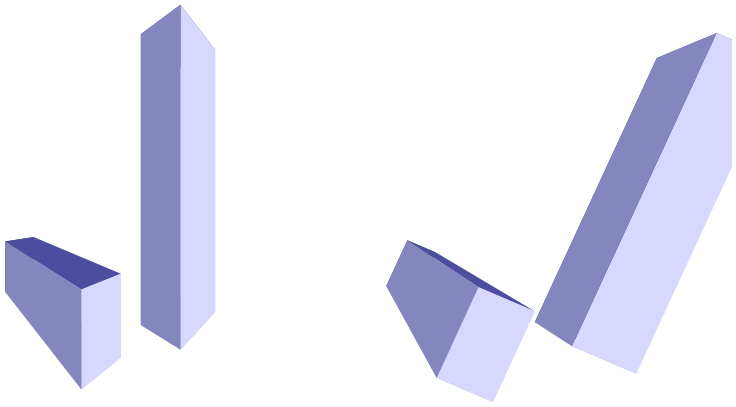


Figure 1.1: Example of an easy Correspondence Problem.

Then, if the local coordinates of the figures in the two images are known, the spatial reconstruction of the scene is straightforward. Moreover, the order between the objects is preserved. Such simple arrangements in which parts unambiguously appear will not always be replicated. The order conservation does not happen in general. Finally, if part of the object falls outside the image, in this case it does not affect the solution of the correspondence. However, natural scenes are likely to be complex and the identification of the same parts of the scene in different images can be a challenging task. This is where the correspondence problem comes into action. The complexity and the usefulness of the applications of this problem has made it one of the most studied among vision questions. In years of research many solutions have been proposed and they perform well in particular situations; however some other problems can be found where the same algorithm performs poorly. Moreover, many solutions rely on assumptions that are sometimes challenged by real-life situation. I refer to the second chapter for a list of such hypotheses. As it is shown in Figure 1.2, the complexity of the plant structure makes the identification of the same corresponding parts difficult. The relative position of the branches changes significantly as well as distances and other

Figure 1.2: Example of vine structure.

geometrical properties. This makes the recognition of so many self-similar branches between the images a complex task. In this thesis I propose new statistical methods that overcome some of the problems occurring in other solutions, while also taking into account the particular structure of the vines.

1.2.1 Automated Pruning Machine

From the point of view of applications, the motivation for my work is to design an automated pruning system, depicted in figure 1.3. The Automated Pruning Project has been active for several years at the University of Canterbury. In its entirety it is a very complex project. A diagram showing its computer vision system core steps is provided in figure 1.4.

The machinery is equipped with three cameras in different positions which take pictures of the same plant from three different poses, as the robot moves at a constant speed. An example can be seen in figure 1.5. The collected images are then independently processed. Wires, posts and the background are segmented out. After this, the 2D structure of the vines is extracted, as in figure 1.6, and information on the connectivity of the structure passed on for the following steps. Given the structure extracted from the images, the correspondence between branches in the images is sought, as well as

Figure 1.3: Picture of the Automated Pruning Machine.

assignments between the 2D and the 3D structure. This is the fundamental step to a correct 3D reconstruction of the vines, like the one pictured in figure 1.7. The 3D model will then be used by the robot arm to cut the right branches, according to a developed AI algorithm.

In the research for the suitable matching algorithm for this problem, there are few requirements to take into account. They are given by the physical characteristics of the machine and the time spent on each grapevine. The dimensions of the robot imply that the 3D reconstruction of the vines has to be included in a given volume. This translates in a constraint for the matching algorithms. The time spent on each grapevine, approximately 2 or 3 minutes, also translates in the choice of the final method used to correspond vines. If a procedure is too slow, it must be rejected due to the discussed time constraint. The last constraint to be considered is that the machine cannot be further modified, so that no cameras of any type can be added. Therefore, the chosen algorithm has to produce the best possible results, given the original apparatus of the machinery. As the Automated Pruning Project is still in a development phase, by *best possible result*, it is to be meant that I am looking for the method providing the largest improvement in terms of number of correct correspondences and topologically correct reconstructed structure over the current matching algorithm.

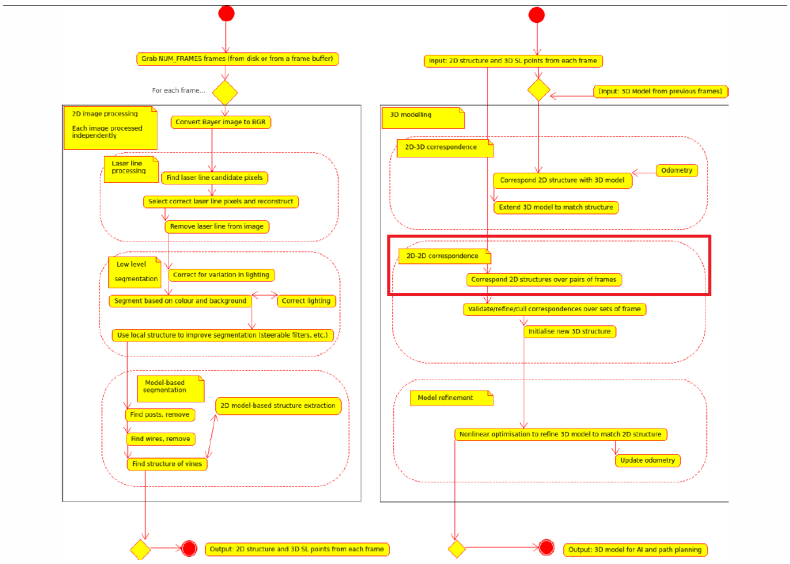


Figure 1.4: Diagram of the vision system of the pruning machine. The red rectangle underlines my specific research as part of the project.

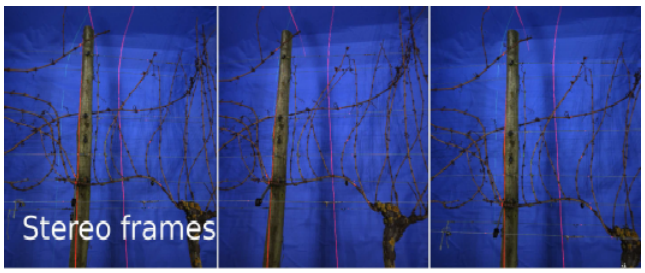


Figure 1.5: An example of the three stereo pictures of a vine, taken by the machine.

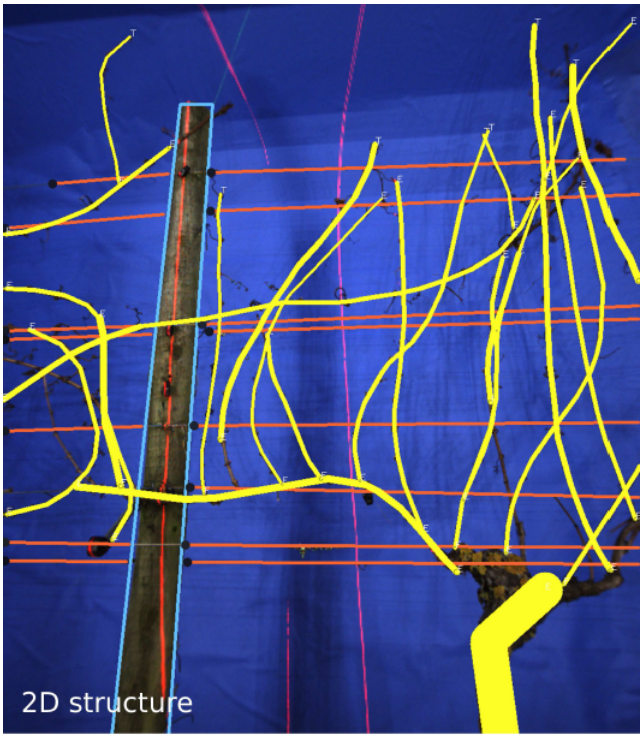


Figure 1.6: Example of an extracted 2D structure of a vine. The letters associated to the branches are used to classify the endpoints of the canes. "T" means that the endpoint is a "tip", that is, the natural end of a branch. "F" stands for "fork": the cane is branching out from another one. Finally, "E" is for "edge", that is the branch is ending on the edge of the image.

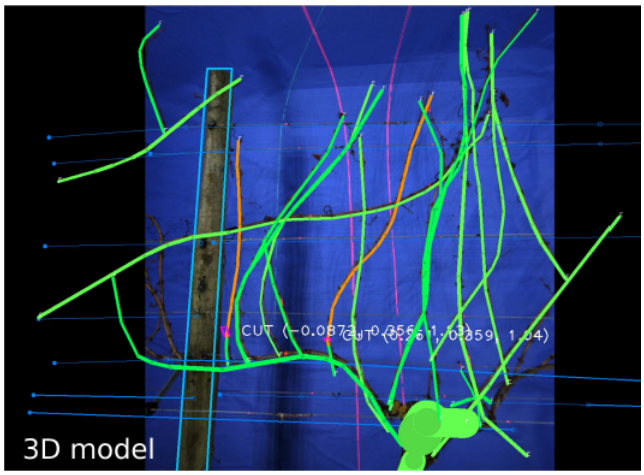


Figure 1.7: Example of a correctly reconstructed vine and then back-projected onto one of the images.

1.3 Statement of the problem

With the term "correspondence problem" a wide number of questions is indicated and many definitions of it are given, depending on the specific issue one wants to address. Generally, the considered problem shares intersections with many disciplines: image analysis, statistics and classification. The common aspect in all the meanings of correspondence problems is the identification of instances coming from the same population.

In the case I treat, the aim is to understand how to match features in different images that represent the same real object, which is the same as classifying the pixels, or the features, as representing the same object. In mathematical terms, a possible definition of my problem is as follows. A surface $S \subset \mathbb{R}^3$ is given. S is then projected onto two different rectangular sets, L and R, of $\mathbb{R}^2$. A set of features of S is available for each of the sets L and R. I call them $\mathbb{F}_L$ and $\mathbb{F}_R$. Solving the correspondence problem, then, is defined as finding a map $M : F_L \to F_R$, with $F_L \subset \mathbb{F}_L$ and $F_R \subset \mathbb{F}_R$, such that a certain loss function $L(M|\Theta)$ is minimized. Here, the generic parameter Θ represents the set of constraints and knowledge available for the problem. In my specific problem, the surface S is a complex tree-shaped

2D manifold.

In the literature, other definitions of the same problem are found. In the case of Object Oriented Data Analysis, Wang and Marron intend with correspondence problem the matching of "parts of one object with the corresponding parts of other members of the population" [153, p. 1851]. Their discussion falls in the field of shape classification and they state that a direct solution is given by $m-$reps, which is a concise way to represent shapes.

Another definition of correspondence problem can be found in Trouvé, [146], in the field of Pattern Recognition. Suppose that we have a set of patterns $f_1, \ldots, f_n \in \mathcal{P}$ and that a new signal $\tilde{f}$ is observed. By taking into account properties of the model and of the problem, a similarity index $S(f_i, \tilde{f})$ can be defined. Then, based on the maximum of those indices, the new signal $\tilde{f}$ can be classified as coming from one of the classes $f_1, \ldots, f_n$.

We can see a common strategy for the proposal of a solution, in all the cases here considered. First there is the identification of a concise way to retain and express information about the problem, that is, defining a suitable model. All the signals treated in our questions and applications are too complex to be handled entirely as they are. There is, therefore, a compelling need for practical ways of describing, numerically, the signals with a waste of informative components reduced to a minimum. A loss of useful elements is inevitable, when compact ways to treat data are used. Moreover the properties of the models play a fundamental role in the proposal of a solution.

Correspondence problems in computer vision are difficult by themselves. Besides measurement errors, imperfections in cameras and computational limits, there is no guarantee of the existence of matching parts or matchings may not be unique. In the problem that I propose to solve, there are further challenges to be aware of that affect the research for a suitable solution. One important problem is the great ambiguity provided by the tree structure. Trees are self-similar, different parts look the same and a branch could appear different from different viewpoints. Therefore, identifying parts may not be easy. Then the presence of numerous branches provides occlusions and depth discontinuities. An occlusion hides parts of the objects pictured in the images, thus providing non-existence of matchings. Moreover, the segmented images that I use could be affected by errors *a priori*.

In Figures 1.8 and 1.9 there is an example showing why the correspondence problem has further challenges with branched structures. Looking at the two branches indicated by the white arrows, it is difficult even for a human eye to understand how to correspond them. This happens because of their overlapping parts. The yellow arrow points at another difficult correspondence. Again, two canes overlap and it is not clear how they can be corresponded. Therefore, there is the problem of determining what cues can be used for the correspondence. For example, a constraint sometimes used in matching, that is the preservation of the orders in which the objects appear, cannot be used: sometimes it is not clear the order with which the canes appear. Instead, physical features of the vines can be used, like length, thickness or the shape of the branches. This consideration leads me to propose *model-based* methods for correspondences.

1.4 Objectives and contributions of the thesis

In this section I list the main objectives and contributions of my research.

1. The principal objective of this thesis is to develop methods to find solutions for correspondence problems in the case of branched structures as defined in the previous section. In particular, I will propose solutions that will be able to cope with some of the problems occurring with branched structures, such as undetected and occluded canes. Moreover, the resulting methods have to be able to avoid making a correspondence if there is not enough evidence supporting it. This is essential because an incomplete reconstruction, but with correct correspondences is preferable to an incorrectly reconstructed vine. I will discuss the achievement of this aim at the end of the thesis. The problem I will study is a new instance of the class of correspondence problems that has hardly been investigated in the previous literature.

2. Since the literature review in the specific problem I investigate is scarce, I carried out an extensive research of more general methods to solve matching tasks that could be applied to my case. Therefore, I believe

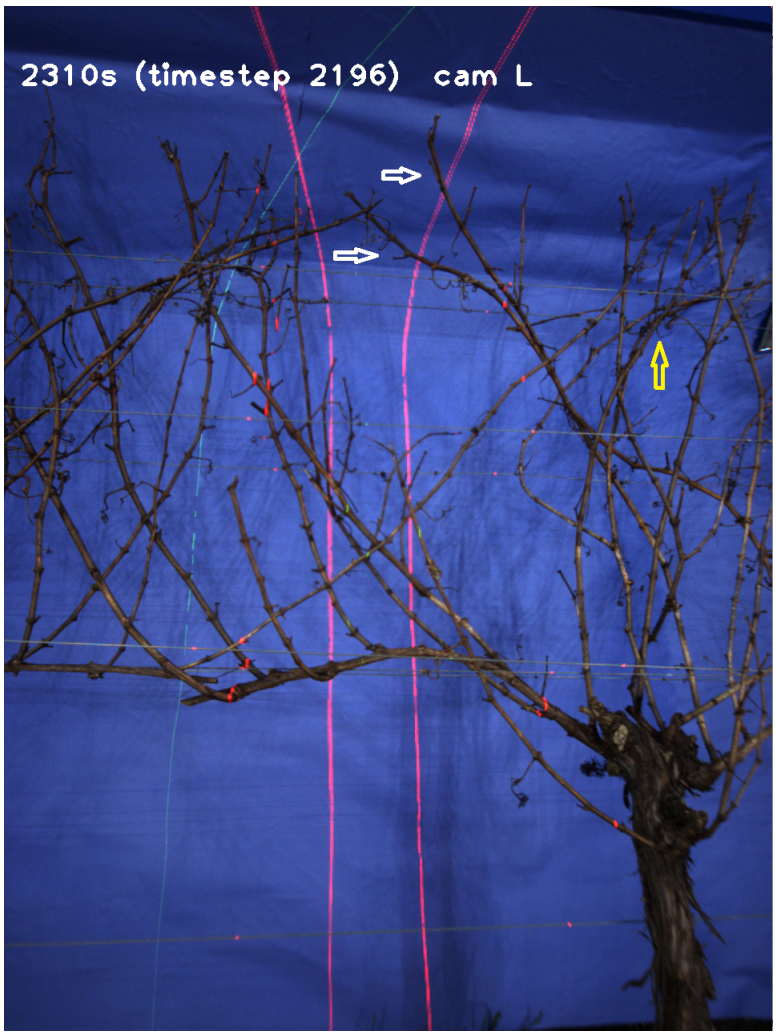


Figure 1.8: Left image of a stereo pair. The arrows indicate hard examples of correspondences. Even for a human eye it is ambiguous and hard to determine the right matchings between the branches indicated. Indeed, canes can overlap for a consistent part of their length and this makes understanding the structure and, consequently, finding the correspondences, difficult.

Figure 1.9: Corresponding right image of Figure 1.8.

that the literature review presented in the following could be used as reference for future research. This is the topic of the next chapter.

3. To find an accurate mathematical model of tree-shaped structures that will be used to select the correct correspondences. I will define variables that take into account geometrical and topological aspects of the structures. I will discuss models for the proposed quantities. Subsequently, I will use the largest set of significant information, from those quantities, to propose models linking the variables that are able to recognize and classify correct matches. This contribution will be reached in Chapter 3.

4. I will develop different methods to solve my problems and compare them on several performance measures. The choice of methods depend on how the informations about variables describing branched structure are used: directly or building more complex models linking them. A thorough comparison of these methods will be performed, in order to choose the best method to be installed on the pruning machine. This objective will be reached throughout Chapter 4,5 and 6.

5. I will design suitable algorithms, written in C++ programming language and with my own code, to perform the methods proposed. The algorithms will compute the solution and will reconstruct the three dimensional structure of the pictured vines. The algorithms are ready to be used on the machine and are integrated with the rest of the project. The algorithms also have to satisfy the time constraint discussed in Section 1.2.1.

The proposed methods will be compared and the best will be installed on the actual pruning robot. The reconstructed vine structure will be used to locate the branches in 3D space. Finally, the AI system of the machinery will decide which branches to cut.

1.5 Structure of the thesis

This thesis will be structured as follows. After this introductory chapter, explaining the motivations, problems and objectives of my research, I will summarize previous works aiming at solving similar problems.

In chapter 3, I will discuss how the vines are modelled, the distributions of the variables learnt and how such information is used.

Chapter 4 will deal with a first group of methods used to address my problem.

In chapter 5, I will present three more state-of-the-art methods that I modified in order to apply to my problem.

In chapter 6, I will present results and finally draw conclusion from my research.

Chapter II

Previous approaches

2.1 Introduction

In the current chapter, I present and discuss previous methods to estimate a three-dimensional structure from a given set of images. The recovering of the actual structure of a scene, using only two dimensional representations of it, is a central task in computer vision. A large number of methods has already been proposed to address the previous tasks. Each procedure relies on assumptions on the scene or on the type of formations of the images.

The methods I decided to discuss here, focus on relevant procedures that have been used to other matching problems or 3D reconstruction tasks. The list of treated approaches begins with stereo correspondence methods estimating disparity maps and discussions of some modifications such as sparse or dense correspondence. After that, I explain 3D reconstruction algorithms. Then, I discuss surface representation. In a natural way, then, I move to statistical methods on manifolds, to end with a brief overview on prior work on shape analysis. To conclude, I discuss some Lie Groups methods and data association problems.

I will give a formal definition of branched structure in the next chapter, but, intuitively, it can be thought of as a tree structure. Only in few cases matching problems with branched structures have been treated in previous literature and the purpose of those works were different from mine. Moreover, often, in plant applications, knowledge of the underlying structure is not fully exploited. On the contrary, methods from recent literature attempt to solve matching problems by exploiting as much as possible the underlying structure of their particular problem. A survey of papers in the "matching" sections of four recent conferences (ICCV 2011, ICCV 2013, IVCNZ 2013 and IVCNZ

Own Alg.	RANSAC	Disparity Stereo Match.	Point Cloud
35	16	13	6
50%	∼22.8%	∼18.5%	∼8.7%

Table 2.1: Frequency and percentage of most used methods in the matching sections of four recent international and national conferences on Computer Vision.

2014) is summarised in Table 2.1. I considered 4 classes of methods:

1. **Own Algorithm**: the authors developed their own algorithm to find matches between images;

2. **RANSAC**: a robust method to fit a model to a set of data;

3. **Disparity Stereo Matching**: a method that looks for correspondences between images by estimating a disparity map between pixels or features of the images;

4. **Point Cloud Methods**: methods exploiting information from depth cameras to estimate the three dimensionality of scenes.

In total, I found 70 papers specifically dealing with matching. In the majority of those, the authors developed their own methods to solve the specific problem they considered. Table 2.2 provides a brief description of approaches to matching problems, presented at ICCV 2013. In general, new customized methods are proposed to better exploit the underlying structure of the problem or because the images were taken with non-common technologies, e.g. aerial photography. Except for the graph matching framework, the methods reported in Table 2.2 are all different and generally tailored on the specific problem. Even in the case of graph matching, some details are customised in order to take into account specific properties of the problem. The second most used method is RANSAC or modifications of it, e.g. [43], [30], [123], [48],[75], [65] or [53]. RANSAC is still widely used and very often present when new matching algorithms have to be compared to the state-of-the-art procedures. It is often used when a set of correspondences has to be estimated together with the camera pose ($\sim$ 70% of the papers). Other works

regarding RANSAC are devolved to extend its use incorporating knowledge of the problems addressed. Only in Raguram *et al.* [123], using a modification of RANSAC, an application with self-similar structures in architecture is shown. Methods using point clouds, such as [148], [119], [66], [160] or [8], also present many adaptations to the specific problem analysed and it is the standard approach when depth cameras are used. Especially the recent work by Ummenhofer and Brox [148] would be useful for the pruning machine; however it exploits an approach using point clouds from depth cameras and this technology is not integrated with the pruning system. Finally, the research in stereo matching focuses on two main directions: incorporating new techniques or relaxing hypotheses in the algorithms ($\sim$ 75% of the papers in ICCV conferences) and developing stereo algorithms with non-standard technologies (the remaining 25%).

Ref.	Problem	Proposed Method	Remarks
[5]	data assignment problem in multi-frame tracking	dual decomposition with Max Product Linear Programming message passing algorithm	better solutions than previous methods.
[21]	matching with points out of focus	local signal equalization	outperforms a state-of-the-art approaches and heuristics applied to the same problem.

[26]	Sparse feature matching	graph matching with density local estimator	Improvements up to 67% on graph matching because it takes into account models and densities of matches.
[29]	Object Matching	graph matching	Improvements of up to 10% on graph matching thanks to a graph learning step.
[42]	matching for image classification	customized-kernel SVM classifier	results comparable to state of the art.
[46]	matching patches	Depth Coherency Sensitive Hashing	results comparable to state of the art, with runtime growing linearly.
[70]	matching patches	regularize disparity map through PatchMatch algorithm	outperforms traditional algorithms in subpixel disparity estimation.
[71]	pattern matching	matching by tone mapping	results comparable to state of the art, but faster algorithm thanks to knowledge of the structure of the problem.

[94]	matching with large displacements	sparse-to-dense matching method for motion field estimation and occlusion detection	results comparable to state-of-the-art.
[95]	feature matching	BRISK	matching BRISK descriptors when scarse *a priori* knowledge of camera pose and scene is available.
[97]	content based image matching	novel Earth Mover's Distance methodology	promising performances by exploiting the specific structure of the problem.
[98]	aerial 3D reconstruction of urban scenes	constrained optimization algorithm	performs better than general stereo matching because of knowledge of the scenes.
[99]	image matching	matching newly defined topological descriptors	useful approach if images are distorted according to a certain class of bounded deformations.

[114]	image matching	decision tree fields	better results than MRF models thanks to the ability of modelling more complex interactions in the structure of the images.
[126]	shape matching	quadratic assignment with elastic net constraint	the method allows the author to obtain sparse-to-dense solutions with accuracy comparable to state of the art.
[132]	matching images with 3D data	3D Nearest Neighbour	state-of-the-art performances on challenging data.
[154]	optical flow computation	novel algorithm tailored on the application	competitive performance together with the ability of managing large displacements.
[159]	multiple graph matching	alternating optimization	competitive results on several datasets, together with a number of advantages over previous methods.
[163]	graph matching	learning parameters and graph at the same time	outperforms conventional methods for learning graph matching.
[170]	dense motion field estimation	robust feature-based matching	comparable to state-of-the-art.

[171]	disparity map refinement in stereo matching	weighted median filter	accuracy comparable to state-of-the-art with faster running time.

Table 2.2: Brief summary of some newly proposed algorithm for matching in ICCV 2013. The works here reported are those specifically dealing with matching tasks. The methods here outlined remark how problem-related knowledge usually brings better solutions.

What can be observed from Table 2.2 is that recent research usually adopts approaches tailored for the specific problem. This, in turn, contributes to achieving results better or similar to some general state-of-the-art procedures. Even in case of graph matching approaches, which is the most frequent framework present in Table 2.2, adaptations to specific tasks provide better solutions.

2.2 Stereo correspondence through disparity maps

In this first section I present some stereo correspondence methods that compute a disparity map to find correspondences between images. Generally, they can be further classified in sparse or dense correspondence and local or global methods. As stated by Szeliski [141, p. 535] "Stereo matching is the process of taking two or more images and estimating a 3D model of the scene by finding matching pixels in the images and converting their 2D positions into 3D depths". Stereo matching is one of the most investigated problem in computer vision. While the physics and geometry relating two or more views of the same scene are well understood, finding correspondences is still a difficult problem. Two different views of a same scene have differences in depth perceptions of the positions of objects, exactly as it happens for the left and right human eyes. The difference in depth induces a *disparity* in the images and this disparity is what is generally sought for to find the correspondences and to estimate the depth of the scene. When it comes to

solve correspondence problems, this is one of the most commonly used way to solve them.

There is a large number of search techniques used to match pixels between different images, but, in the case of stereo matching, there is more information that can be exploited, i.e. the positions of the cameras taking the images and calibration data. Thanks to this, through epipolar geometry, the number of matching candidate pixels can be reduced, because matching points have to be found on corresponding epipolar lines. Two or more views induce an intrinsic geometry, depending only on the cameras' parameters, which provides strict constraints that matching parts of the images have to satisfy. For example, given a point x in one image, epipolar geometry provides an epipolar line, on another image, where the appropriate point matching x has to lie.

Generally, a common treatment before any stereo algorithm is the *rectification* of the images. Rectification is a process of the images that ensures that corresponding epipolar lines are horizontal and parallel to scanlines. Moreover, it corrects distortions and make the disparity for points at infinity to be null. As a result a *standard rectified geometry* is obtained and it is used in a large number of stereo methods. This new geometry gives a simple relation between 3D depths and disparities. If the 3D depth is called Z and the disparity d, the rectified geometry provides the relation

$$d = f\frac{B}{Z}, \tag{2.1}$$

where f is the focal length and B is the distance between two views, or baseline. Equation (2.1) states that the disparity is inversely proportional to the distance from the observer and, clearly, directly proportional to the distance between the two views. Therefore, the relation between corresponding pixels in two images is given by

$$x' = x + d(x, y) \text{ and } y' = y. \tag{2.2}$$

Thanks to this treatment, estimating the depth becomes a matter of estimating the disparity map $d(x, y)$.

Suppose two rectified images L and R are given. Let $p(x, y)$ be a pixel in L, lying on the y scanline and let $q(x', y')$ be the corresponding pixel in R. Because of the rectification process, $y = y'$ and $x' = x + d$, where d is the disparity map between the images. Then, the general procedure for estimating d is to minimize the following functional

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d). \tag{2.3}$$

The first term, $E_{data}(d)$ is the *data term*, representing a similarity measure between the pixels, whereas the second term, also called *spatial term*, $E_{smooth}(d)$, is a penalization term used to have a smooth solution, because d is supposed to be piecewise smooth on surfaces. λ is a tuning parameter, that controls the smoothness constraint.

Several proposals for the data term are known. The most used, according to Sarkis and Diepold [131], thanks to the simplicity of evaluating them, were the absolute difference or the squared difference of the appearance of the pixels:

$$C(x, y, d) = |p(x, y) - q(x + d, y)|$$

or

$$C(x, y, d) = (p(x, y) - q(x + d, y))^2.$$

Since the former is usually more robust, it is now preferred to the latter. Usually, to make the data term more robust against occlusions or intensity variations, see, e.g. [141], [16] or [131], a truncated version of E_{data} is used, namely

$$E_{data} = \min\{C(x, y, d), k\}, \tag{2.4}$$

where k is a threshold value. Similarly for the smoothness term, a common choice is

$$E_{smooth} = \min\{|d(x) - d(x + 1)|, \tau\}, \tag{2.5}$$

where τ is another threshold value for the spatial term.

Stereo algorithms are widely used to find correspondences between images. However, in general as observed by Szeliski [141] or Birchfield and Tomasi [18], there are still some pitfalls. I report a list of common assump-

tions and problems, taken from Birchfield and Tomasi, [18].

1. Each point in the scene looks identical in the images. This is also referred to as "hypothesis of Lambertian surfaces". An obvious example where this does not hold is reflection.

2. The depth of the scene is piecewise constant in the direction of the scanlines.

3. At every depth discontinuity, there is an intensity gradient between the near object and the far one.

4. Every object contains a modest amount of intensity variation. This is supposed because, otherwise, untextured surfaces cannot be reconstructed.

5. Variations on *ordering constraints:* the pixels to be associated have the same relative location, no scene point is viewed from both sides of another scene point, if an object is partially visible from one view, then it must happen also for the other ...

6. Problems in managing specularities and occlusions and pre-processing of the images.

7. Matching ambiguities.

While some assumptions are reasonable, others do not hold in general. For example 4. and 5. are often challenged and the particular structure of the scene can complicate further the framework. However, part of the recent literature on stereo matching is devolved to overcome these problems.

2.2.1 Sparse correspondence

Historically, the first stereo correspondence methods where *feature-based* or *sparse*. They extracted a set of features in the two images and then tried to match them using suitable metrics or similarity measures. There are three good reasons for choosing this kind of algorithm (Szeliski [141])

1. it is not computationally expensive;

2. a limited number of features provides more correct matches;

3. if images with different illumination are to be matched, features matching is more stable.

Recently, sparse stereo algorithm have been used as seeds or initial steps to generate further matches, Szeliski [141], Sarkis and Diepold [131].

In Figure 2.1 there are two images used for a sparse correspondence method. The feature extraction step produces the images shown in Figure 2.2. Then, the computation of the functional E in equation (2.3) gives the disparity map showed in Figure 2.3. An example of a sparse stereo

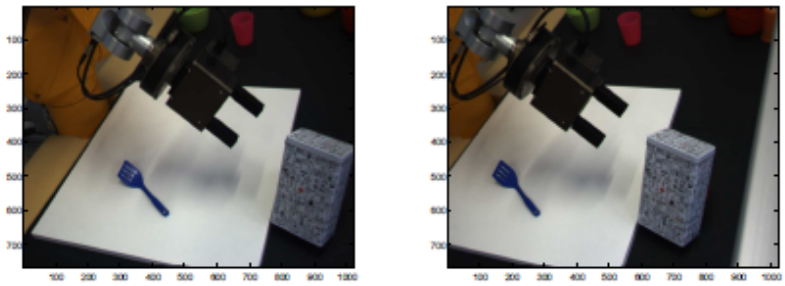


Figure 2.1: Example of a sparse correspondence method. These are the input left and right images.

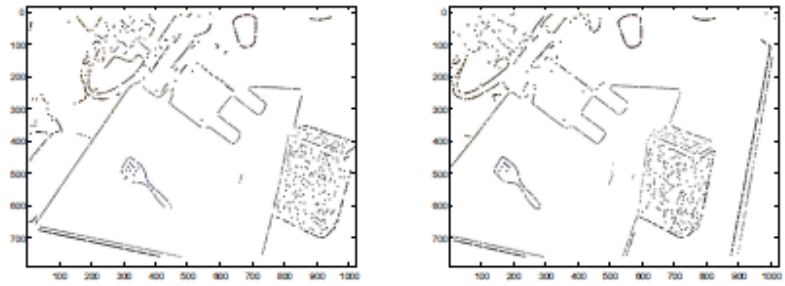


Figure 2.2: Feature-extraction step.

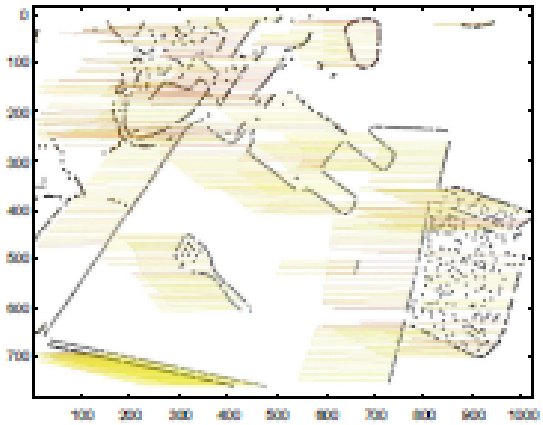


Figure 2.3: Sparse disparity map. All the images taken from Jessen [76].

algorithm is provided in Sarkis and Diepold [131].

The method proposed by Sarkis and Diepold [131] is a standard sparse stereo matching, where the cost function C is the absolute value of the difference in the appearance of the pixels. However, they extracts features only from one image, for example, L, keeping the other, R, dense. This choice is due to the way they decided to find the minimum of the functional E, through a sparse belief propagation (BP) algorithm. The motivations for using their modification of a BP algorithm are of a computational nature: less memory required, more precision and speed (Sarkis and Diepold [131]) compared to standard BP algorithms. The disparity map obtained in this way is sparse. To recover a dense disparity map, the authors appeal to a weighted average of the sparse disparity map. Precisely, for every set of three points forming a triangle, the quantity $\hat{d} = \sum_{i=1}^{3} \omega_i d(x_i)$, where ω_i are weights to be computed and satisfying suitable geometrical constraints.

Another example of sparse correspondence is the matching of profile curves or occluding contours, as described in Cipolla and Blake [32]. The shape and location of curves vary together with the camera centres. So, when curves are used to recover the depth of a scene, erroneous reconstructions are frequent. However, if the images are very close to each other, for example, the cameras are close or the camera is moving slowly, through the fitting of circular arcs it is possible to recover immediately surfaces in the

space. The fitted arcs enable obtaining a sparse surface mesh directly from the matches. Cipolla and Blake [32] use this method for robotics applications. They demonstrate how robots can circumnavigate curved obstacles or pick up curved objects thanks to their method.

2.2.2 Dense correspondence

Other stereo algorithms used nowadays rely on a dense correspondence. In this case the problem may be even more challenging than sparse correspondence, especially in inferring depths of surfaces without any texture. Dense stereo algorithms can be further classified in local or global methods, depending on the particular optimization step they solve. In this section I briefly discuss them.

Dense stereo algorithms follow the approach already described of minimizing the E functional in (2.3). Contrary to sparse methods, there is no feature-extraction step. The disparity map is computed using the entire rectified images. What mainly distinguish dense stereo algorithms is the choice of the similarity or dissimilarity measure to be used in the functional minimization and the way the minimization is solved.

Figure 2.4: Example of two input images for a dense stereo algorithm. Szeliski [141].

In Figure 2.5 the estimated disparity map coming from the images in Figure 2.4 is shown. Dense stereo algorithms rely on assumptions on the observed scene, (Scharstein and Szeliski [135], Szeliski [141]). Nonetheless,

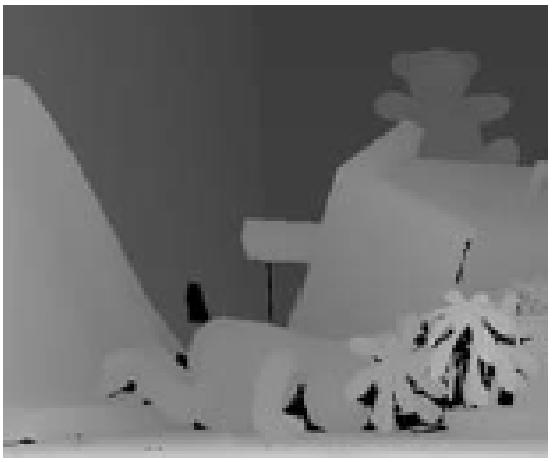


Figure 2.5: Computed dense disparity map. Szeliski [141].

dense stereo algorithms are known to perform well in many situations.

A recently proposed approach to stereo matching is Efficient Large-Scale stereo matching (ELAS), proposed by Geiger *et al.* [57]. In ELAS, a probability for the matchings is built from a set of robust corresponded points, called "support points". The first step in the method is to find the support points. These points are pixels that, thanks to their texture, uniqueness and other consistency criteria, can be unambiguously matched. Let $S = \{s_1, \ldots, s_M\}$ be the set of support points. The image coordinates of the s_i's are then used to create a mesh through Delaunay triangulation. The correspondence of the observed points inside the mesh will then depend on S. Therefore, indicating with d the disparity map at a pixel and o^l and o^r, respectively, an observed point in the left and right image, the joint distribution can be written as (assuming the left image is the reference image)

$$P(d, o^l, o^r, S) \propto P(d|o^l, S)P(o^r|o^l, d).$$

For the prior $P(d|o^l, S)$, the authors give an expression which is a combination of a uniform and a Gaussian distribution, with parameters depending on the support points. For the image likelihood, a constrained Laplace distribution

is proposed. The maximum of the probability is then computed, minimizing the corresponding energy functional. Geiger *et al.* state that the experiments show the precision of their algorithm is comparable to the state-of-the-art procedures and faster.

However, even a method like ELAS, when applied to branched structures, does not provide good enough results for a final aim like that of the pruning machine. In the work by Ni and Burks [113], an ELAS algorithm is adopted for a 3D reconstruction of a tree canopy. The aim of their research is to measure overall variables of the plants, such as height, width, volume and leaf coverage. As it can be observed in the Figures 2.6 and 2.7, and even

Figure 2.6: Left and right images of a small lemon tree as used in [113].

more in other examples in [113], the reconstruction of the structure is good enough for their purpose, but parts of the branches are missing and some of the boundaries are not well defined. When the leaves are close enough that the depth discontinuity is not large, the reconstruction is acceptable. With my images, instead, the depth discontinuities are more marked and the thinness of the branches makes the problem more difficult for this type of method.

2.2.3 Local methods

Local dense stereo methods minimize the functional E in (2.3) on neighbourhoods of pixels or on suitable defined windows of pixels, Szeliski [141]. A criterion to determine the right size of the window is needed. It has to

Figure 2.7: Reconstruction of the lemon tree seen in Figure 2.6.

be large enough to contain textures, but small enough not to come across depth discontinuities. In this way local disparity maps are produced. So, there is the need to aggregate them to have a globally defined disparity map. Moreover, the local disparity maps have to be aggregated taking into account the smoothness constraints. Almost every method aggregates disparity maps using convolutions methods. The major source of difference in local dense stereo algorithms is in the way the different disparity maps are aggregated together.

2.2.4 Global methods

In global dense stereo algorithms, a disparity map on the entire image is obtained and there is no need for an aggregation step, since the smoothness term performs a similar function. Even for this subgroup of methods, many techniques for the optimization have been proposed. The relations between these methods and Bayesian or random field frameworks have been shown

and investigated, for example in Scharstein and Pal [134]. Optimization procedures include dynamic programming, simulated annealing and, more recently, max-flow and graph cut methods.

2.2.5 Final comments on stereo matching methods computing disparity maps

The hypotheses assumed in stereo matching algorithms exploiting disparity maps make them unsuitable for my problem. The prior probabilities used in this class of methods are suitable for scenes with "smooth" changes of depth. Instead, the structure of the vines pose a completely different scenario: the surfaces to be matched are thin and the depth discontinuities are frequent and often sharp. This class of methods seeks correspondences by minimizing energy functionals that rely on similarity or dissimilarity measures on the pixels. However, the structure of the vines, being self-similar, makes the use of similarity measures unsuitable: it generates too many ambiguous admissible matches. This also implies that a pixel-to-pixel matching method in my problem is not the most appropriate.

2.3 Algorithms used in stereo correspondence

In this section I briefly present some of the most common algorithms used for solving stereo correspondences. It could happen that the method used to compute the best stereo matching leads towards the choice of determined algorithms. It could also happen the converse: a proposed algorithm, exploiting specific information about the problem, influences the functional used in stereo matching. Here I explain graph cuts, belief propagation and dynamic programming.

Graph cut algorithms have been used in many fields of computer vision, including segmentation and stereo matching. The graph cut algorithm is also known as min-cut and, as shown for example in Wu and Leahy [157], it is equivalent to max-flow. I follow the approach explained in Kolmogorov and Zabih [88]. Given a weighted graph $G = (V, E)$, it can be partitioned into disjoint sets by removing edges connecting the parts. A degree of dissimilarity between the parts of the partition can be computed. It is the total weight of the edges removed. In graph theory this is called a *cut*. For example, if a

graph is partitioned into two subgraphs, A and B,

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v), \tag{2.6}$$

where $w(u, v)$ is the weight of the edge connecting the nodes u and v. The definition of cut is used in the general *labelling problem* and the stereo matching problem can be formulated in terms of a pixel labelling problem. The goal is to assign to each pixel $p \in P$ a label from a set $L = \{f_1, \ldots, f_m\}$. In stereo problems the set L is just the set of disparities. Thus, in the stereo matching problem the target is to find the labelling $f = \left(f_1, \ldots, f_{|P|}\right)$ that minimizes:

$$E = \sum_{p} D_p(f_p) + \sum_{(p,q) \in N} V(f_p, f_q), \tag{2.7}$$

where N is the set of neighbouring pixels. D_p is the penalty for assigning a label to a pixel p and V is the penalty for assigning a pair of labels to neighbouring pixels. In stereo matching, the label f_p will be assigned to the pixel p if the pixel p in an image I corresponds to the pixel $p + f_p$ in another image I'. If some conditions in the problem are met (I refer again to Kolmogorov and Zabih [88]), the minimization problem in (2.7) can be solved with a single graph cut. However, any minimization problem involving the Potts model (the term with V in (2.7)), is NP-hard, therefore, the best possible achievable result, as stated by Kolmogorov and Zabih [88], is a local minimum.

Another often used algorithm to find stereo matching is belief propagation (BP). I refer to Yedidia *et al.* [161] for a detailed description of the algorithm. BP can be used to estimate marginal probabilities or most likely states, e.g. the Maximum A Posteriori (MAP). BP starts from the observation that often joint probabilities factorize into more simple factors:

$$P(x_1, \ldots, x_4) = C \cdot f(x_1, x_2) g(x_2) h(x_2, x_3, x_4). \tag{2.8}$$

The factors, also called *potentials*, are not in general probabilities. However, in some cases, for instance Markov chains or Bayes networks, they can be interpreted as conditional probabilities. As an example, in Figure 2.8, the

joint probability of the Bayes network is given by

$$P(x, y, z, w) = P(w)P(x)P(y|w)P(z|w, x)$$

and, in Figure 2.9, the joint probability of the chain is given by

$$P(x, y, z, w) = C \cdot f_{wx}(w, x)f_{xz}(x, z)f_{yz}(y, z)f_{wy}(w, y).$$

BP has been proved to provide an exact solution in case there are no loops

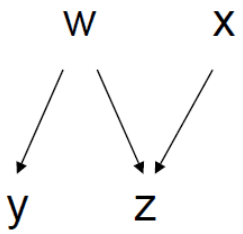


Figure 2.8: Bayes network example.

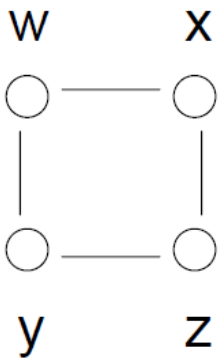


Figure 2.9: Markov Random Field example.

in the graph, e.g. chains or trees. BP is an iterative process in which neighbour variables X_i and X_j pass *messages* $m_{ij}(X_j)$. Each of these messages

represents how the node X_i believes the marginal value $P(X_j)$ to be. The initialization of the algorithm is usually setting all the initial messages to 1 or assigning them positive random values. Then at each iteration, the messages are updated with the usual laws of probability and eventually convergence is reached. The convergence is computed, for example, comparing successive steps of the algorithm. As an example, I consider an application to a MRF stereo algorithm. Suppose an unknown disparity field $D(x, y)$ is to be estimated. Let us take as prior $P(D) = \frac{1}{Z}e^{-\beta V(D)}$, where $V(D) = \sum_{r,s}|D(x_r, y_r) - D(x_s, y_s)|$ sums over all neighbouring pixels. Now, let $M(D) = |R(x + D, y) - L(x, y)|$ denote the matching error between the images and let the likelihood function $P(M|D)$ be $\frac{1}{Z'}e^{-\mu M(D)}$. Assume conditional independence: $P(M|D) = \prod_{(x,y)} P(M(D(x, y))|D(x, y))$. Then the posterior is found through Bayes' theorem $P(D|M) = P(D)P(M|D)/P(M)$ and an application of BP provides an estimation of the marginals at each pixel.

The last method I explain is dynamic programming. Dynamic programming, rather than being an algorithm itself, is an algorithmic paradigm. It is a procedure in which complex problems are divided into smaller and simpler tasks, whose answers are then used for the initial problem. A well-known example is the branch and bound algorithm in combinatorial optimization. If we want to find the minimum (or maximum) of a function f over a set S, we can partition the set S into smaller sets $S_1, \ldots, S_n$ and note that $\min_S f = \min\{m_1, \ldots, m_n\}$, where $m_i = \min_{S_i} f$ (branching step). We have now n usually simpler problems. However, we can compute upper and lower bounds for the desired minimum value (bounding step). Then, if we know, e.g that $\min_{S_i} f \geq \max_{S_j} f$, the set S_i can be surely disregarded when looking for the minimum of f. Dynamic programming is used in many fields: mathematical optimization, shortest path problems on graphs, computer vision, statistics and many other disciplines. One of the most common applications of dynamic programming in stereo matching is the following. Since matching pixels are found on corresponding epipolar lines, the energy functional in (2.3) or (2.7) can be decomposed as the sum on simpler energy functionals on the different scanlines. This optimization can be performed efficiently using dynamic programming. However, it has been noticed that this ap-

proach generates vertical inconsistencies among the scanlines. To overcome this problem, modifications of energy minimization on scanlines have been proposed. As an example in Veksler [151], algorithms taking into account neighbours of pixels have been considered.

2.4 Graph matching and tree matching

Graph matching is an essential problem in many applied scientific fields, such as computer vision, pattern recognition or machine learning. As Table 2.2 has shown, graph matching approaches are becoming more and more used to address the problem of finding correspondences among images. In general terms, the graph matching problem is to determine a mapping between the nodes of two graphs, such that the mutual relationships between the nodes of the graphs are preserved as much as possible. With the development of combinatorial optimization, graph matching has been framed in an Integer Quadratic Programming (IQP) setting. Graph matching looks for correct correspondences between two attributed graphs $G^P = \left(V^P, E^P, A^P\right)$ and $G^Q = \left(V^Q, E^Q, A^Q\right)$. V represents the set of nodes, E the edges and A the attributes. In feature correspondence, $a \in A$ describes the local appearance of a feature in an image. Given a pair of nodes $\left(v_i^P, v_j^P\right)$, $\left(v_a^Q, v_b^Q\right)$, in the two images and the relative edges e_{ij}^P and e_{ab}^Q, a similarity or compatibility, that measures the goodness of matching between the given nodes and edges, can be defined as $\mathbf{W}_{ia;jb} = f(a_i^P, a_j^P, a_{ij}^P, a_a^Q, a_b^Q, a_{ab}^Q)$. The compatibility $\mathbf{W}_{ia;jb}$ is computed considering both the similarity of nodes and edges and the mutual consistency. Usually, a set of correspondences is represented through a permutation matrix $\mathbf{X} \in \{0,1\}^{n^P \times n^Q}$, with $X_{ia} = 1$ if the node v_i^P corresponds to v_a^Q, with n^p and n^Q being the number of nodes in G^P and G^Q respectively. Indicated with $\mathbf{x} \in \{0,1\}^{n^P n^Q}$ a column-wise replica of $\mathbf{X}$, the matching problem can be reformulated in terms of IQP as

$$\mathbf{x}^* = \arg\max(\mathbf{x}^T \mathbf{W} \mathbf{x}) \tag{2.9}$$

$$\text{s.t. } \mathbf{x} \in \{0,1\}^{n^P n^Q}, \forall i \sum_{a=1}^{n^Q} \mathbf{x}_{ia} \leq 1, \ \forall a \sum_{i=1}^{n^P} \mathbf{x}_{ia} \leq 1.$$

The two-way constraints encode the one-to-one matching constraint. The

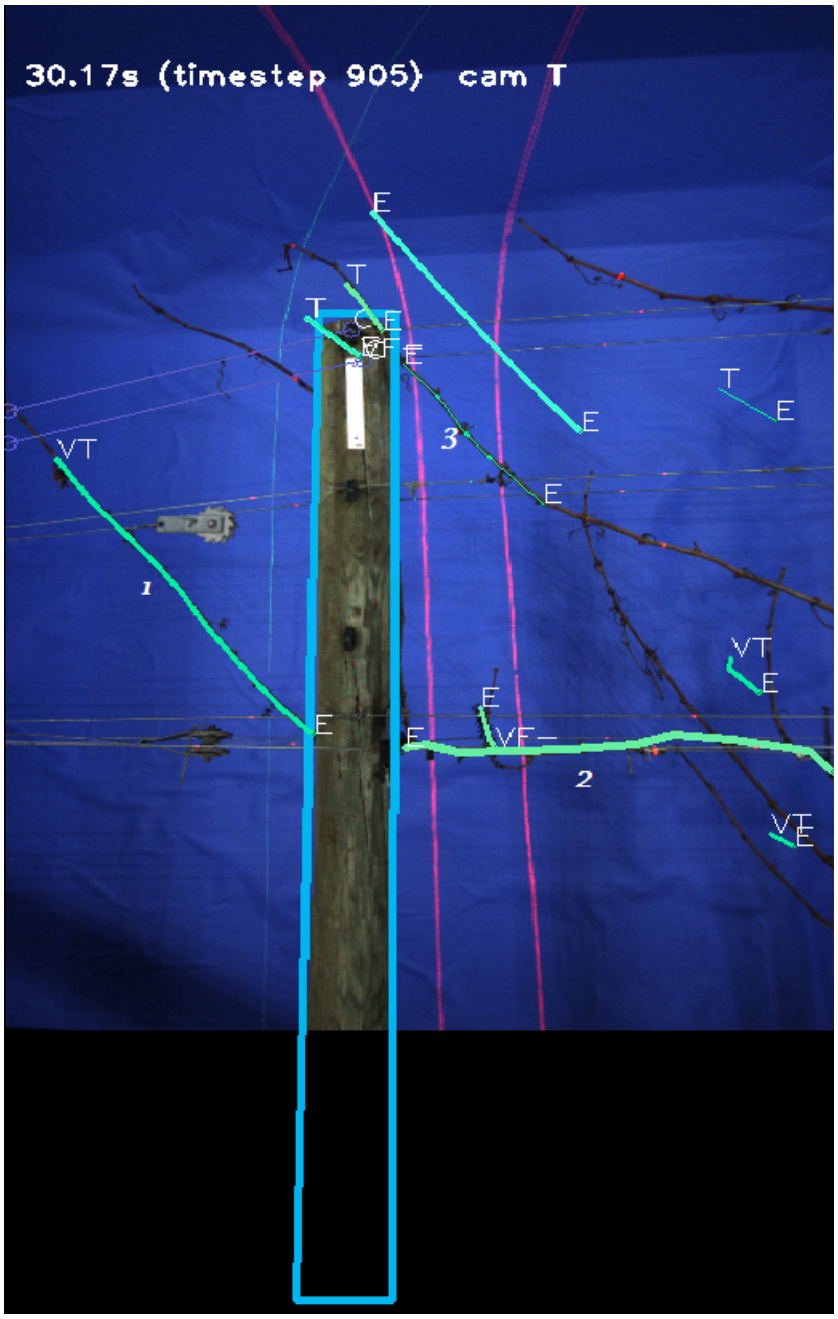


Figure 2.10: Example of selected correspondences after the application of a Graph Matching algorithm. The letters have the same meaning as in Figure 1.6: E stands for edge, T is for tip and F for fork. They correspond to how the structure extraction algorithm automatically interprets the endpoints of the branches.

branched structures I deal with can naturally be encoded as a graph. However, it is a specific subset of graphs among those investigated in the graph matching framework. The problems in graph matching usually assume complete graphs, i.e. the nodes are all connected. Instead, a vine has a *tree* structure. A tree structure provides a strong matching constraint based on the ancestry relationships between the nodes. Both graph and tree matching are NP-hard problems. Exploiting the tree structure in a graph matching framework simplifies the problem. I will describe in detail a tree matching algorithm for my application in Chapter 4. It follows a similar approach to Cho *et al.* [27] and Kumar *et al.* [91]. In Figure 2.10 there is a preliminary application of a tree matching approach to my problem. It can be seen that three branches, numbered 1,2 and 3, are correctly matched and backprojected in the image. There are other branches that are wrongly matched and reconstructed, as their backprojection does not overlap with any actual branch or the structure. An explanation can probably be that matching through similarity measures is not the best choice with self-similar structures.

2.5 General feature-based methods

General feature-based methods exploit an approach similar to sparse correspondence to estimate a 3D reconstruction of a scene. Some relevant features are extracted from the images and then the correspondence is done starting by matching the sets of extracted features. However, they do not necessarily seek for the estimation of a disparity map. In this section I briefly discuss about feature-based methods like the ones by Dey *et al.* [39] and Delaere *et al.* [38].

Dey *et al.* [39] use a featured-based correspondence method to propose a 3D reconstruction of vine plants. A number of images of vine plants is collected and a dense three dimensional point cloud is generated in such a way that it captures the overall shape of the scene. The main goal is to classify which of the points obtained belong to grapes, foliage or branches and infer the crop yield. The authors use a support vector machine to classify points. The information the classifier relies on are mostly given by shape features. Even though the reconstruction is sufficiently accurate for their

aim, the structure is too inaccurate, because of too many ambiguous matchings. This flaw could be reduced if an appropriate model of the vine is taken into account. An example of application of the algorithm in [39] is shown in Figure 2.11.

In Delaere *et al.* [38] the aim is the development of a computational

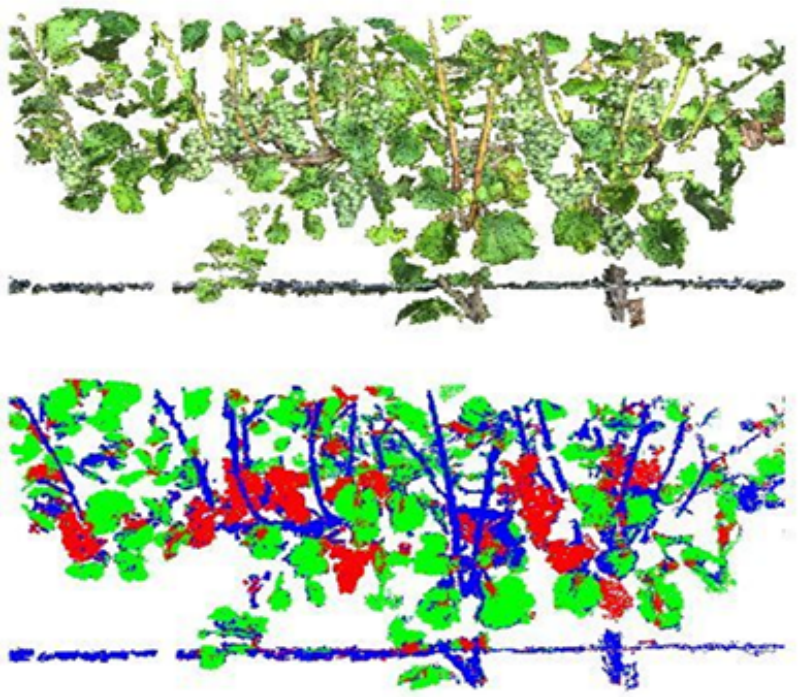


Figure 2.11: Illustration of grapevine reconstruction in [39]. Many missing branches can be noticed. The top figure is the reconstruction obtained, while the bottom one is the colour-coded classification of the plants.

vision system for the automatic interpretation of blood vessels on multiple angiographic projections. The focus is on the solution of two practical problems: the reconstruction of the three dimensional structure of blood vessels and the automatic localization of the coronary atherosclerotic disease. See Figure 2.12 for an example of two images representing the angiographic projections of the real structure. The method used for the recovery of the structure is simply based on the constraint given by the epipolar geometry

and on the setting of a suitable statistic defining the goodness of matching. The authors use a heuristic method to solve their tasks. The epipolar geometry permits only to associate lines to points. Therefore there is the need of a complementary criterion to find the right matching points. The authors in [38] construct a statistic, indicated with H, aiming to measure the goodness of matching, by measuring the physical properties of the blood vessels represented on the two images such that the statistic H is high if the physical characteristics of the points are similar.

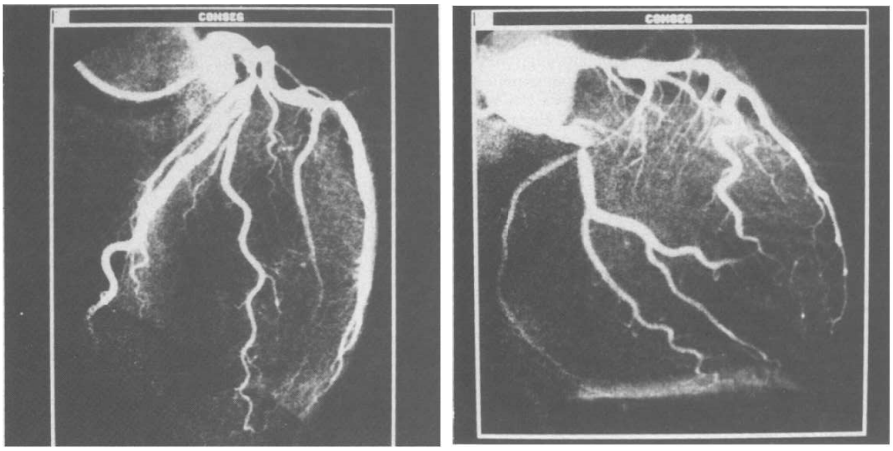


Figure 2.12: Example of blood vessels to be reconstructed. Source [38]

In this section, I also review the work by Teng *et al.* [142]. Teng *et al.* aim at reconstructing tree trunks for computer graphics, starting with two slightly different stereo images of the same tree. Besides the difference in the purpose of their problem compared to mine, there are several other aspects that render their research not applicable to my case. Firstly, their structure extraction step is obtained through a user interaction iterative process, in which initial seeds for a segmentation are manually selected. The authors assess that this can be time consuming, but the results are good. This structure extraction process is different from the same step in the prun-

ing machine. Indeed, the manual choice of seeds can reduce ambiguity, in their case, in the 2D segmented structure and therefore can result in less errors. This is not achievable, instead, on our pruning application and, as I showed in the previous chapter, Figures 1.8 and 1.9, the structure is ambiguous even for human eyes. Once the 2D structure is recovered in the two images, the corresponding points between the views have to be found. Teng *et al.* do not explain the method they used to solve the correspondence problem. It is only understood that they do not use knowledge of the tree structure, but minimize, in a least-square sense, a cost function. The camera matrices are known, as well as the fundamental matrix. Therefore, the authors parametrize points in the images as projections of unknown 3D points and limit the search for corresponding points along epipolar lines. This method alone cannot guarantee a good solution in general. So, in [142], a minimum curvature constraint is imposed, in order to have a smooth 3D branched structure. Usually, this search for correspondences based on points, breaks down in the presence of occlusions and the authors are aware of this flaw. This method seems to work well in their case. However, the number of branches they have is not high and the self-similarity between branches is not as marked as in grapevines. As it can be seen in Figure 2.13, the results

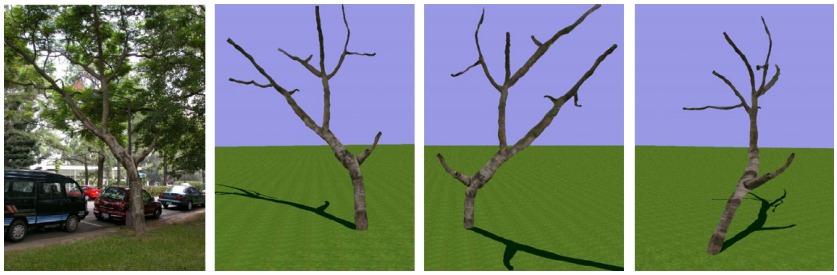


Figure 2.13: Example of a tree reconstruction. Source [142]

are good for computer graphics. However, if the reconstruction were to be used for the pruning machine, it would not be as useful. For example, in the last image in figure 2.13, proportions are not consistent and the tree itself is distorted. This is likely to influence the 3D localization of the tree. So, there would be a possibility, for the robot arm of the pruning machine, to

completely miss the plant or to collide with it.

2.6 3D reconstruction

Stereo algorithms are not the only possible technique used to estimate a three-dimensional scene, given some images of it. There is a large number of techniques exploiting different features of images. For example, shape and orientation of surfaces can be estimated using shades or textures. In this section I briefly describe some methods that do not seek for correspondences to estimate the 3D structure of a scene. In particular, I discuss voxel carving, surface representation and model-based reconstruction methods.

2.6.1 Voxel carving

Voxel carving is a technique used to reconstruct the volume of an object given multiple views of that object against a background. First, a voxel grid is built around the object we are interested in reconstructing. Clearly, not all of those voxels can be accepted. It can be possible that, when a voxel is projected on one of the images, it lies outside the boundaries of the object. Therefore, a voxel, according to Bălan [9], is defined to be *volume consistent* if its projections inside the cameras are inside the silhouettes, or boundaries, of the object. Then, intuitively, a good reconstruction of the object is made of all the volume consistent voxels. This idea can be made more mathematical. The projected region of a voxel v inside camera k is denoted by $Proj^k(v)$ and by S^k the boundaries of the object as seen through camera k. If V is the volume of the object we want to reconstruct, then

$$v \in V \Leftrightarrow Proj^k(v) \subseteq S^k,\ \forall k.$$

Finally, as the projection of a voxel can occupy several pixels, a voxel is set to be volume consistent if a proportion of its projections lies inside the silhouette: $|Proj^k(v) \cap S^k| \geq \varepsilon |Proj^k(v)|$, with $\varepsilon < 1$.

I review an application of voxel carving to recover a three dimensional network structure. Zheng *et al.*, [166], aim at reconstructing a three dimensional structure of roots, given multiple two dimensional images of the scene.

The method adopted is a slight variation of voxel carving and makes use of the new concept of regularized visual hull. First, they divide their method into three steps. This is done because the authors try to overcome problems deriving from the sensitivity to image quality and calibration. The first step is the modelling of the background through harmonic functions. This allows the authors to improve the extractions, through adaptive threshold, of the silhouette of the roots in each image. Let $k = 1, \ldots, N$ be a generic image. Following [166], I indicate with F_k the k-th foreground, U is a set of voxels in the three dimensional space, $\pi_k(U)$ is its projection onto the k-th image and $\pi_k^{-1}(F_k)$ is the maximal set of voxels with projection F_k. The *visual hull* is the maximal set of voxels such that its projection is contained in all the foregrounds F_k's:

$$V = \bigcap_{k=1}^{N} \pi_k^{-1}(F_k). \tag{2.10}$$

Or, alternatively, using the definition modified by Zheng *et al.*, [166], the visual hull can be viewed as the solution of an optimization problem. The *consistency* of a voxel v, with the image k, is defined as the function

$$cons_k(v) = \begin{cases} 1 & \text{if } v \in \pi_k^{-1}(F_k) \\ -N & \text{otherwise,} \end{cases} \tag{2.11}$$

where N is the number of images. Then the visual hull is the set of voxels maximizing the total consistency:

$$V = \arg\max_S \sum_{v \in S} \sum_{k=1}^{N} \mathrm{cons}_k(v). \tag{2.12}$$

However, this last criterion is not suitable for the purpose illustrated in [166], as even little calibration errors can cause inconsistencies between the images such that the three dimensional back projection is almost empty. Therefore, the authors suggest choosing an image I_j, among the ones at disposal and to use it as a benchmark image. Then the *regularized visual hull* is defined as:

$$V = \arg\max_S \sum_{v \in S} \sum_{k=1}^{N} \mathrm{cons}_k(v) + \lambda |\pi_j(S) \cap F_j|, \tag{2.13}$$

where $|\cdot|$ denotes cardinality and λ is a tuning parameter. The role of the regularization term is "to encourage" the voxels to cover a specific area F_j. In other words, this is equivalent to assuming that the image j is a guide in the reconstruction process. The three dimensional reconstruction is therefore a compromise between two objectives: a good global approximation and a more definite structure, made precise by the choice of a single image. Finally, the solution found in (2.13) could be made only of many different connected components, therefore a final step has to be added to the method: a regularizing step to repair the global connectivity. Zheng *et al.* [166], compare the use of the regularized visual hull to the use of only the simple visual hull in their method. The regularized visual hull performs better as it increases the true positive ratio (defined as the number of covered silhouette pixels divided by the total number of silhouette pixels), with only a slight increase in the false positive ratio (defined as the number of covered pixels not in the silhouette divided by the total number of silhouette pixels).

The voxel carving approach just described has been proven useful for reconstructing branched structures. However, it requires a larger number of images, for each single structure, than what the pruning machine produces. Moreover, even small calibration errors between the views can generate important problems propagating throughout the system. A voxel carving algorithm is used, anyway, to model the trunk of the vines. It is used by the collision detector of the robot arm to compute the volume of the trunks and also a collision region. Even if it is not very accurate, it is well suited for its purpose, because it allows to compute a larger volume around the trunk, therefore assuring the robot arm does not collide with it.

2.6.2 Surface representation and interpolation

One of the most adopted techniques to reconstruct a surface is to interpolate it starting from a set of sparse data point. This method is similar to a curve fitting. Given some data points known to belong to a surface, an analytic expression of a two-dimensional surface is sought, in such a way that it approximates well the data. Some smoothness constraints are to be considered, in order to have good solutions. Both parametric and non-parametric

methods have been proposed and investigated to solve this kind of problem. In any case, some topological assumptions have to be made. I briefly report an example of this technique, as developed by Ettinger *et al.* [47]. This is related to vines, because each branch can be seen as a 2D surface in the space, such as the surface in Figure 2.14.

Ettinger *et al.* in [47] address the problem of reconstructing a two dimensional surface of an artery embedded in $\mathbb{R}^3$. The authors need to reconstruct a single artery with an aneurism to estimate the shear stress acting on that surface. In the reconstruction process the smooth and nonlinear nature of the surface has to be taken into account. The general problem is, given some sample points over the surface, estimate a function f recovering the real surface. It is posed as a penalized minimization problem, where the penalization is suitably chosen in such a way that the solution function is sufficiently smooth. The functional to be minimized is

$$\sum_{i=1}^{n}(w_i - f(\mathbf{u}_i))^2 + \lambda \int_{\Omega}(\Delta f)^2 d\mathbf{u}, \tag{2.14}$$

where w_i's are the sample points, Ω is the domain of the function, λ is a tuning parameter and Δf is the Laplacian of f. It is known, in general, how to solve (2.14) on planar domains. The non-planar nature of the points w_i's is, now, to be taken into account. Therefore, conformal maps can be defined such that they map the domain Ω onto a Euclidean space. This results in considering a new penalization term in (2.14), given by the Laplace-Beltrami operator Δ_Σ on Σ:

$$\lambda \int_{\Sigma}(\Delta_\Sigma f)^2 d\mathbf{u}, \tag{2.15}$$

where Σ is now the image of Ω through the conformal map used. The solution is then obtained with a finite elements method, where the mesh grid adapts itself to the geometry of the blood vessel. After that, the function f can be recovered as well as the three dimensional structure of the blood vessel. In Figure 2.14 the reconstructed surface obtained with the method just described, is shown.

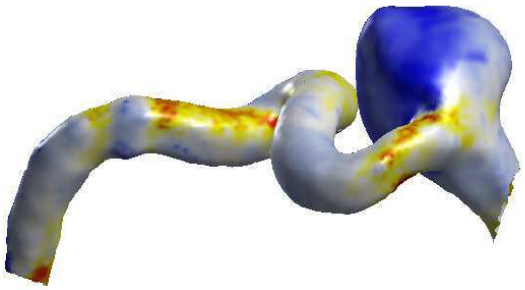


Figure 2.14: Reconstructed surface obtained with the method proposed by Ettinger *et al.* [47]. The different colours correspond to the different shear stresses acting on the surface.

2.6.3 Model-based reconstruction

In particular applications, some information about the three dimensional appearance of objects are known in advance; this provides strong constraints for the proposed reconstruction. For example, in architecture, buildings or objects are made of simple geometrical regions or parametric shapes. This kind of technique is also frequently used to model heads and faces.

An example of an application in architecture is provided by Debevec *et al.* [37]. They proposed a method combining a geometric modelling with a *model-based* stereo matching. In the first step, the geometric modelling, the authors choose block elements and align their edges with visible edges in the input images. Debevec *et al.* state that this step is more reliable than general feature-based structure from motion, because, in architecture, geometry plays a fundamental role. Once the geometry has been estimated, a stereo algorithm is used. In this case, it is called model-based because the knowledge of the geometry provides stronger constraints for the correspondences. This is the case even for vines: a model-based approach can eliminate from the beginning a large number of not-admissible matches; moreover, knowledge of the structure of the vines provides a strong constraint for correspondences.

In the case of heads and faces modelling, a similar approach is used. Principal Component Analysis (PCA) is used to obtain a low-dimensional

representation of faces. This simplifies the problem and allows to design a model-based algorithm that exploits directly the smaller set of parameters found through PCA, Szeliski [141], Mumford and Desolneux [109].

2.7 Statistics on manifolds methods

In this section I discuss some methods that naturally come from a generalization of two ideas already explored in the previous section. Some procedure have been proposed to recover the shape of a surface or to reduce the dimensionality of complex "sets" like the set of faces. Could anything similar be proposed for more general sets or structures? Could a general shape in some complex space be recovered from a sparse set of data? Could an analogue of PCA be defined in these spaces? Some investigations addressing the previous questions have been undertaken and some useful methods have been proposed. They proved to be very helpful especially in the field of medical imaging. In this section I explore some of these methods. This group of methods is linked to my application in that a vine plant can be seen, as already said above, as a two-dimensional manifold embedded in the space. Therefore, similar techniques or ideas could be used in my application, for example to define suitable statistics that take into account the non-linear structures of these surfaces.

A generalization of PCA is provided by Fletcher *et al.* [52]. Their aim is to describe the variability of a population of anatomical structures, starting from a set of images. The classical method to approach this task is exactly PCA. The problem is how to generalize this in order to describe the variability of a population of complex structures. Fletcher *et al.* want to describe the variability of a population of hippocampi. A hippocampus could be seen as a manifold in a three-dimensional space. Therefore, the authors need a synthetic way to represent it. They first compute the medial axis of the hippocampus, defined by the locus of the centre of the spheres inscribed in it. Then, for each centre the normal vectors pointing to suitably representative tangent planes and the radius of the sphere are computed. So, for each centre there is a 4-tuple $\mathbf{m} = \{\mathbf{x}, r, \mathbf{n}_1, \mathbf{n}_2\}$. These 4-tuples are called *medial-atoms* and this is the synthetic representation of the hippocampi, as shown in

Figure 2.15: medial-atoms are sampled among and within the hippocampi. Now, the data points do not lie in a Euclidean space, so standard statistical techniques do not hold any more. Even the basic concepts, such as mean or distance have to be redefined. I refer to Pennec [121] for a detailed discussion of the topic. However, the set of medial-atoms is locally diffeomorphic to a Euclidean space, so that a concept of distance can be defined.

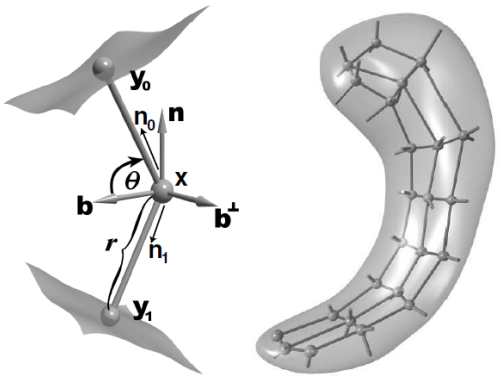


Figure 2.15: Medial-atom and medial-atom representation of an hippocampus. Image taken from Fletcher *et al.* [52].

The concept of geodesic replaces distance. Then the standard procedure to obtain the principal components is carried on using geodesics. Fletcher *et al.* call this new technique Principal Geodesic Analysis (PGA). A description of the variability of the population is then obtained in a lower dimensional space. In the future the authors aim to apply this new technique to image segmentation.

A similar approach has been used by Joshi *et al.* [79] to analyze the shape variability of human kidneys and to segment them in applications to medical imaging. They demonstrate the efficacy of their method even in case of simulated training data.

A different approach to the similar problems is to linearize the spaces involved in the analysis. Examples of such a technique are provided in Vaillant *et al.* [149] or Begelfor and Werman [12]. I briefly explore the former of the

two cited works.

Since the work started by Grenander [62], a deformable template can be thought of as an orbit under a group of diffeomorphisms. In this way, the comparison between objects is transferred to a comparison of the diffeomorphisms generating them. This approach can be used to match objects: if there exists a path between two diffeomorphisms, generating two different templates, that minimizes a certain measure, then the two templates can be matched. This framework proved to be very helpful for the comparison of anatomical objects. Vaillant *et al.* [149] use a property, called "conservation of momentum", which the minimizing paths enjoy, to model the diffeomorphisms. This approach allows a dimensionality reduction of the problem and the translation of the considered framework to a linear space. This permits the author to derive a simpler criterion for feature matching. Finally this method is applied to landmark matching on a set of hippocampi and faces.

2.8 Other general statistical methods

In this section I review some methods that are purely statistical in their nature, but do not strictly belong to the previous denomination.

The closest method to the ones of the previous section is the one proposed by Wang and Marron [153] on Object Oriented Data Analysis (OODA). The development of their new method is motivated by a statistical analysis of a set of tree-structured objects, namely blood vessels. Figure 2.16 provides an example of the structures analysed in [153]. Wang and Marron want to study the variability of a population of blood vessels. In order to do this, another generalization of PCA is needed on a space where the elements are trees. Again, the key concept is a suitable notion of distance on the space. In this case, a metric which takes into account the topology of the trees and the presence of observed features is designed. Having done this step, the classical notions in statistics such as average tree, variance or principal components can be translated in the space of trees. The application investigated by Wang and Marron [153] is close to mine in the structure of the data, but the target is different. They are interested in a study of the distribution of trees. Nonetheless, their method is useful as an example of statistical techniques

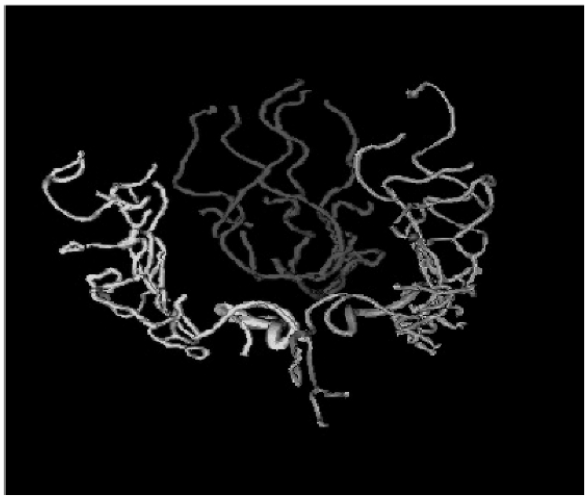


Figure 2.16: Example of a datum analyzed in [153]. The structure here represented is very similar to vine plants.

applied to branched structures. Wang and Marron design a suitable distance on the space of tree-shaped structures that can be decomposed in a term considering only the topology of the tree and in another term considering attributes of the trees. A suitable consideration of their analysis could be very useful in defining concepts relating to the topology of the vines.

An approach closer to the estimation of the disparity map is the one proposed by Belhumeur and Mumford in [16]. Their aim is to find a method for estimating the disparity map, between two views, that is also able to handle occlusions. As already seen in (2.1), for each point x, the disparity between two images I_l and I_r and the depth are related by the relation

$$Z(x) = f\frac{B}{d(x)}.$$

So, they want to find the $Z(x)$ such that $P(Z(x)|I_l, I_r)$ is maximized. Thanks to Bayes' theorem $P(Z(x)|I_l, I_r) = cost \cdot P(I_l, I_r|Z(x))P(Z(x))$. Now, the approach focuses on finding the right expression for the two probabilities. The term $P(I_l, I_r|Z(x))$ is the one that gives the probability of the images given the depth map. This term takes into account the possibility of occlusions, by

counting the points not mutually visible from the cameras. For the prior, they rely on a Markov Random Field hypothesis. Belhumeur and Mumford first discretize the depth function Z and then they assume the following dependency: $P(Z_{k+1}|Z_k)$. This MRF hypothesis for general scenes is unrealistic for several ways: mainly because the dependency on the nearest neighbour pixels is too restrictive and because there are usually more nonlinear interactions between objects. However, they give a reasonable expression for the prior, differentiating, as seen in Figure 2.17, the three cases $Z_{k+1} \prec Z_k$, $Z_{k+1} \succ Z_k$ and $Z_{k+1} \approx Z_k$. These events are used to describe the presence of obstacles occluding parts of background objects. Finally, an algorithm for finding the maximum of the posterior probability is given, but it relies on another unrealistic assumption: the ordering constraint. I consider the method proposed by Belhumeur and Mumford [16] to be very interesting and able to provide a deeper insight in the occlusions problem. The strength of their method is to estimate the correspondence and the non-matched regions simultaneously. Indeed, "depth estimates, occluding contours and half-occluded regions are inseparably linked" [13, p. 433].

The last two methods in this section exploit the presence of invariants in projective geometry. Images are naturally identifiable with projective planes and it is well known that there are some quantities that are invariant for projective transformations. The works of Maybank and Beardsley [105] and Rothwell *et al.* [127] follow this last approach. Both works aim at recognizing objects by computing some of the invariant quantities in observed objects and comparing them to template values. In the former work, only the cross-ratio is used; in the latter more general invariants are considered. Maybank and Beardsley are even able to derive an analytical expression for the probability density function of the cross ratio. This also allows for the computation of confidence intervals.

2.9 Shape space methods

A recent development in the field of classification is represented by *shape analysis*. Studying the shape of observed objects or of a set of points can provide useful information for recognition purposes or to define probabilistic

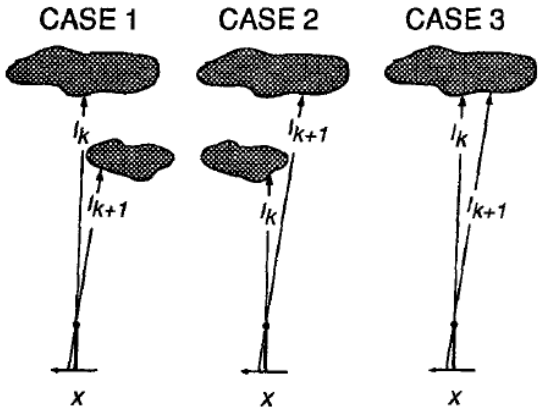


Figure 2.17: The three cases taken into account for the prior term in [16]. In the first case, if the distance Z_{k+1} is less than Z_k, it is expected to have an obstacle at the right of the current point Z_k. Conversely for case 2. Finally, if the depths of two contiguous points are comparable, no obstacle is expected between them. Image and discussion taken from [16].

models. First of all, a rigorous definition of the term shape is needed. This is not an easy task. The same objects can have different shapes if we look at them from different points of view. Moreover the "set of shapes" is intrinsically infinite-dimensional: there is no finite set of features able to represent a shape in a unique way. The previous sentence means that if a finite set of variables could be extracted to describe the shape of objects, then it is always possible to find two different objects that, nevertheless, are represented by the same variables. Finally, there is no possibility for a "set of shapes" to enjoy a general linear structure. In the current section, I review two different definitions of shape and their applications to recognition and classification.

The first approach to a definition of shape, is provided by Kendall [86]. The author, in his application, is interested in defining the shape of a configuration of points. Intuitively, the shape should be a concept which remains invariant if rotations, translations or isotropic dilatations are applied to the configuration of points. This is exactly what Kendall does: he defines the shape as a quotient space. He considers k points in $\mathbb{R}^m$. First, he quotients

out the dilatation group, replacing every point z with $z/\|z\|$, thus obtaining the unit sphere $S_m^k \simeq S^{m(k-1)-1}$. Finally, the rotations are quotiented out:

$$\Sigma_m^k = S_m^k / SO(m),$$

where $SO(m)$ is the group of rotations in $\mathbb{R}^m$. The elements of Σ_m^k are equivalence classes and each of them is a shape. This resulting space is a finite dimensional manifold, which admits a representation for low dimensions. For example, the shape space of all triangles on a plane is identifiable with the sphere S^2 in $\mathbb{R}^3$ with radius 1/2. A shape space can be endowed with a metric, which is the essential tool for statistics. Examples of applications of such a method are found in astronomy, geography or archeology. Kendall [86] himself applies this analysis to classify human bones in an archeological application and to astronomy to validate a theory on the distribution of quasars.

In case of points in a finite-dimensional vectorial space, if the dimensions involved are not too high, representations of shapes can be found, as for the set of triangles on a plane. However, how can a concept of shape be defined for real objects? How can similarity between shapes be measured? Sharon and Mumford address these questions in [138]. The idea of the two authors is to assign to every shape a sort of fingerprint and every shape will then be identified with it. Sharon and Mumford define a shape to be a diffeomorphism f of the unit circle S^1 to itself, in such a way that it is increasing and satisfies the relation $f(x+2\pi) = f(x)+2\pi$. The group of the defined diffeomorphism is denoted by $\mathrm{Diff}(S^1)$. This group is finally quotiented by the group of Möbius transformations, that is $z \mapsto (az+b)/(\bar{b}z+\bar{a})$, which is exactly the invariant group of transformation in Projective Geometry. Therefore, the authors are working with elements in the space

$$\mathrm{Diff}(S^1)/PSL_2(\mathbb{R}).$$

This quotient space is then endowed with a metric that makes comparison between shapes achievable.

Ideas from these works are essential as a starting point to analyse the

shape or the topology of vines. The ideas provided in the discussed works can be used or further developed and modified in order to fit to the problem I deal with. For example, notions or suitable variables to analyse the shape of the vines can be designed and statistically modelled. If it is found that the correct correspondences/reconstructions have a particular shape or distribution for their shape, then that would represent a strong constraint to decide which are good matches. A "fingerprint", like the one used by Sharon and Mumford, could be assigned to the shape of the branches and used for the correspondences. I will discuss this in the next chapter.

2.10 Lie group methods

Among the most recent techniques in pattern recognition there is the use of Lie group methods. These procedures share similar aspects to the methods already described of Sharon and Mumford [138] and Vaillant *et al.* [149]. Indeed, methods exploiting Lie Group theory attempt to model the images or shapes they deal with as applications of elements of a Lie group to a benchmark object. Examples of works in this field are given by Miller and Younes [108], Trouvé [146] or Younes [162].

Lie Algebras are vector spaces over a field enjoying additional properties with respect to a binary operation called *Lie bracket*.

The approach they used is the same: the key idea is to model the signals or the images as the action of a group of transformations to some templates. Usually, these transformations belong to a Lie group. Under this view, the works by Sharon and Mumford [138] and Vaillant *et al.* [149] can fit the same framework, although the group action considered, for example in Trouvé [146], is more general. The use of Lie group methods leads to several advantages: the statistical modelling part is now moved to the description of the group actions involved. Moreover, by choosing suitable groups or metrics on the groups, many invariant properties can be taken into account. Since the images are naturally modelled as projective planes and there is a natural group acting on projective planes, the idea is to use Lie groups to relate the two views. This could give an estimation on how the object would appear in an image. Then a matching is provided by maximizing a similarity measure

involving the previous estimation.

The problem of comparing images or recognition tasks become now a matter of confronting the group actions generating the observed images or shapes. The distances defined to achieve such targets involve, usually, the solution of complex differential equations or variational problems, which are solved by standard numerical methods.

Trouvé [146] shows how this approach can be used to solve problems in the fields of pattern classification, template fitting and pattern matching. Younes [162] applies Lie group methods to match planes as seen in different images through different points of view.

2.11 RANSAC and JCBB

The last set of methods to address the correspondence problem that I review, includes RANSAC (RANdom SAmple Consensus) by Fischler and Bolles [51] and the JCBB (Joint Compatibility Branch and Bound) by Neira and Tardós [112]. I briefly describe here both of them.

RANSAC is used to fit a model to a set of data and it is capable of modelling data which incorporate an amount of outliers. This makes it suitable for applications in image analysis, where features, that are subject to errors, are to be matched.

The algorithm as presented in [51] is, basically, as follows. Suppose to have a set of m data points. Select at random n points out of those m, where n is the minimum number of points necessary to generate a model and then build a model using the sampled points. After that, the data points consistent with the model are counted, whereas the others are rejected. This procedure is repeated until a consistent model with the largest set of data points is found. If the model is correct, then the selected data points are inliers. RANSAC is shown to perform well even in case of a large ratio of outliers. See [106] for a more detailed discussion of RANSAC's performances. RANSAC is one of the principal methods used to find correspondences between images. RANSAC has been proven useful when relative pose and inlier matches have to be estimated jointly. Moreover, it is well suited when the matches ambiguities are due to unknown relative pose. I will describe in more

detail in Chapter 4 how RANSAC is applied to my problem. In Figure 2.18, a preliminary application of RANSAC to corresponding branches is shown. Looking at Figure 2.18, it seems that, even if some branches are missed, there

Figure 2.18: Preliminary application of RANSAC to my problem. These are the constructed and backprojected correspondences considered correct with RANSAC. The reconstructions are projected back in the image taken by the Top camera.

is, overall, a good set of corresponded and reconstructed branches. However there is an evident error. For example, it is clear that the branch indicated by the arrow comes from a wrong correspondence.

The other method, the JCBB, is mostly used in navigation systems. It tries to solve a matching problem between sensors within the same tracking system. JCBB can measure the goodness of a set of pairings by rejecting wrong associations. This detail makes the JCBB algorithm more robust even

in case of complex environments. The authors in [112] define a new criterion to verify the correctness of the matchings, based on correlations of errors and the joint compatibility of a set of matching features.

2.12 Conclusion

Figure 2.19: The complicated structure of vine plants is likely to make most of the known methods used in correspondence problems to fail.

In this chapter I have presented some methods that either have already been used to solve problems similar to mine or that present some interesting aspects that can be used for my problem. I have briefly explained stereo matching through disparity maps, surface reconstruction and other statistical methods. However, the problem that I aim to solve poses additional challenges. As it can be seen in Figures 2.19 and 2.20, major issues are represented by the structure of the vine plants. It is self-similar, often non-planar and with frequent occlusions and overlapping. The ambiguity of matching, in the case of vine structures, is often very high. Stereo matching meth-

Figure 2.20: It is evident the frequent presence of self-occlusions and the self-similarity of the structure. Moreover, some branches can overlap for long parts of their entire length, as indicated by the white ellipse.

ods computing disparity maps contain assumptions not holding in my case, inappropriate prior probabilities and poor performances near depth discontinuities and occlusions. Moreover, a pixel-to-pixel approach for the correspondences is not suitable, in my application. Many feature-based methods are not useful either: they fail in presence of self-similarities and they detect features that are not suitable for three-dimensional reconstruction, such as background features. This happens sometimes in the structure extraction step of the pruning machine as well: some features in the background or on the wires are detected as branches, thus introducing another source of error when looking for correspondences between branches. Surface reconstruction or point clouds methods cannot be tested in my case, because the pruning machine is not endowed with high resolution depth cameras. However, it would be interesting to see how those methods would perform with vine structures when affordable very high resolution depth cameras exist. RANSAC in case

of depth discontinuities, self occlusions and repetitive structures, shows reduced performances and the probability of a matching to be an inlier for the model is significantly lowered, as shown by Chum and Matas [31] and Zhang and Kosecka [165]. However, because of the high generality and robustness of RANSAC, it may be that a suitable definition of a good model for the vines could result in a satisfactory application of that method.

Model-based methods could be the most suitable approach to address my problem: they are designed on the particular structure of the problem and rely only on knowledge of it with few extra assumptions. Therefore, the main objectives of following chapters are:

- finding efficient models for the vines and characterizing good reconstructions;
- defining ways to classify correct matches from wrong ones;
- proposing methods using the classification step to find the correct correspondences;
- comparing the proposed methods.

With this chapter I consider achieved the second objective detailed in Section 1.4, namely providing a thorough literature review for correspondence problems in the case of branched structures.

Chapter III

Proposed modelling of branched structures

3.1 Introduction

In the present chapter, I explain how the vines in my problem are modelled. I discuss the type of variables used and some of the learning procedures to estimate distributions and parameters.

An accurate modelling of vines is fundamental for the model-based methods I propose to address the correspondence problem in the case of branched structures. The decision of adopting a model-based strategy to approach my problem derives from an attempt to fully exploit a better knowledge of the structure of the problem and to use the resources provided by the pruning machine.

I summarise again here, why I do not expect the methods presented in the previous chapter to be directly applicable to my problem or to perform well in my case.

- Some of the previously proposed procedures use different technologies, such as point clouds.

- Some methods have assumptions that do not hold in my case or have different aims.

- Many of the presented algorithms show drawbacks with repetitive and self-similar structures.

- Some of the methods have an approach to finding correspondences that is pointwise. However, because of the structure of the vines, a point-to-point correspondence is not suitable for my problem.

A model-based approach, instead, seems more suitable for my problem. First because it shrinks the space of admissible matchings. The knowledge of the structure implies on one side that the reconstructions will be topologically and geometrically correct; on the other side, that the only assumptions introduced deal with the structure itself. A model-based approach allows to be less dependent on similarity or dissimilarity measures to find correspondences and, with the self-similarity of the vines, it is a great advantage. My methods exploit an approach to the correspondences that is "branchwise", that is I correspond branch to branch, rather than point to point. One of the great advantages of the proposed methods is that some problems with occlusions and self-similarity, as well as some of the drawbacks shown by other methods, are significantly reduced.

In this chapter, I begin with a definition and discussion of a mathematical model for the vines and branches that I used, together with an explanation of all the variables used. Subsequently, I present the learning step used to estimate the distributions of the variables and how those information are used to design algorithms seeking for the correspondences.

3.2 Branched structures and model of vines

In this section I present some definitions that will be used throughout the rest of the thesis and a description of vines and their branches from a mathematical point of view. The following definition is inspired by Feragen *et al.* [50].

Definition 3.1. *A branched structure is an embedded tree in* $\mathbb{R}^2$ *or* $\mathbb{R}^3$ *and consists of a series of edge embeddings (branches), connected as determined by a rooted, ordered (combinatorial) tree.*

In Definition 3.1, the term *tree* implies that cycles are not allowed in a branched structure. There are no restrictions in the number of children that a branch can have. For any branch, I will refer to the branch it comes out from as its *parent* or *ancestor*.

A *vine*, therefore, from a mathematical point of view, is simply a branched structure. However, Definition 3.1 is quite general. To specify more the definition of a vine in my case, I assume that a vine is a branched structure

where every branch is either a 2D manifold in $\mathbb{R}^3$ or a 1D manifold or a 2D surface in $\mathbb{R}^2$. In the following I will specify further assumptions on those manifolds. In case of vines, I will refer to the root of the tree also as *trunk* and to the branches also as *canes*. An equivalent representation of a vine is as a graph $G = (V, E)$, where the set of edges E encodes the information about the connectedness of the structure and each node in V is a branch.

The number N of branches for each vine is unknown; therefore it will be represented as a random variable on (strictly) positive integers. The random variables describing the number of branches in different vines are assumed independent.

A discussion about the random generation of branches is more complicated. It seems a natural assumption to consider each branch as the realization of a stochastic process. Given two random vines, they can be considered independent. However, for each vine, the branches depend somehow on the structure. Anyway, as there is no simple visible relation between different branches of the same structure, I will assume that they may depend only on their parent. That is: given two different random canes of the same vine, they are not independent; on the other hand I will assume them as independent *given* their parents. A reason for this assumption is that information about connectedness and mutual relations between branches are taken into account and this gives another constraint when looking for correspondences.

The methods proposed to solve the correspondence problem rely on the knowledge of the three dimensional structure of the vines. A brief description of the common idea behind the methods is the following. Let L and R two images representing different projections of the same vine. I indicate with $|L|$ and $|R|$ the number of branches seen in L and R respectively. Let l_i, $i = 1, \ldots, |L|$ and r_j, $j = 1, \ldots, |R|$ be two generic branches in L and R and let $\omega_{ij} = (l_i, r_j)$ indicate the matching between the $i-$th branch of L and the $j-$th one in R. Then, supposing ω_{ij} satisfies some constraints and a "realistic" 3D reconstruction can be computed from l_i and r_j, it can be considered as an admissible correspondence. In this process, the Levenberg-Marquardt algorithm is used to reconstruct the branch defined by that particular matching. By "realistic" 3D reconstruction, I mean that the selected match does not produce a degenerate reconstruction, it does not fall outside an admis-

sible volume or has some measurements, defined in the following, assuming suitable values according to some distributions. Given the reconstruction and depending on the values assumed by those measurements, some properties about the correspondence and the structure can be inferred, namely the admissibility of a matching; moreover the probability of a match being correct can be computed. It follows that a more efficient method to identify the correct correspondences could be designed, if more accurate information about those variables are known.

In order to have the best representation of vines, then, the set of considered variables should be as thorough as possible. As I defined above a branch to be a manifold embedded in a suitable space, there are three classes of variables that can be taken into account in proposing a suitable model. The sets of variables are:

1. geometrical variables;

2. topological variables;

3. other variables.

I now proceed to a description of the three classes here considered.

3.2.1 Geometrical variables

As the name suggests, the class of geometrical variables deals with geometrical properties of the branches or with quantities that can be computed using only geometrical properties of the branches. In the mathematical idealization, I assume that the manifolds representing the branches are smooth and that they have uniform width along their longest axis. In Figure 3.1 an example of a mathematical model for a branch can be seen. These assumptions seem fair, because in reality, canes do not usually exhibit sudden changes in their growth direction. For what concerns the hypothesis about the width, it is reasonable because the change in width along the branches is small. Moreover, even though a local thickness is observable at any point along the 2D structure of the vines, from a computational point of view it is more convenient dealing with a uniform average width along the branches.

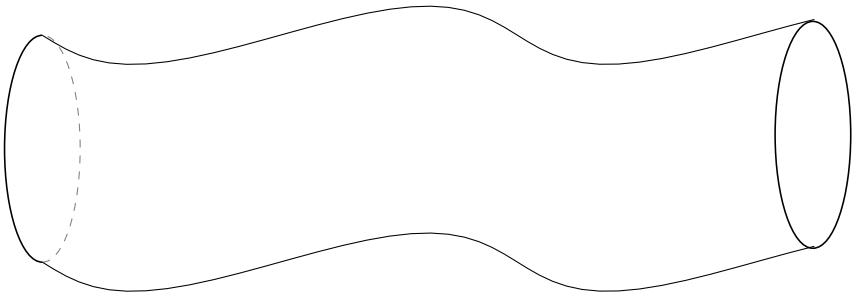


Figure 3.1: Example of a mathematical model for branches: smooth surfaces with constant width.

As a matter of fact, it is better building a model with a uniform thickness, rather than local ones.

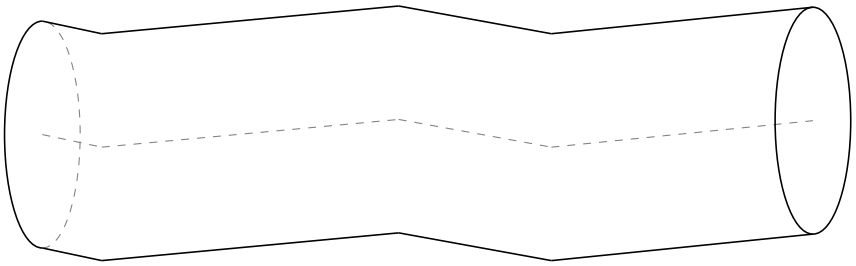


Figure 3.2: Example of a numerical approximation of a mathematical branch adopted by the calculator. The dashed line is the polyline approximation of the centreline of the branch.

As depicted in Figure 3.2, the pruning code approximates the branches with piecewise linear functions, in the following called *polylines*. However, the same branch, in two different images can have a different number of linear approximating functions. Therefore, if the local widths were to be taken into account in the correspondences, it is not clear how to combine them together in the particular 3D branch defined by those matchings. Considering the average thickness gives another simpler representation of branches: a centreline curve or a polyline approximation together with a real positive number describing the average width itself. In general, in defining the variables, I use the properties of the centreline or polyline approximation of the branches. I will refer again to the manifold idealization only for the Euler Characteristic

in the next section. When I am interested in just the polyline, I will indicate a branch as a set of points in $\mathbb{R}^3$, such as $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where n refers to the number of endpoints of the polyline segments. The points $\mathbf{x}_i$ will also be called *control points*.

The first two simple geometric variables considered are, therefore, the *average thickness* of the branch and its *length*, defined as the length of its centreline.

Defining other geometric variables is more complicated. Suppose to have the centreline approximation of a branch, such as the piecewise linear curve in blue in Figure 3.3. The *kink angle at a point* $\mathbf{x}_i$ is defined as the planar angle between the two segments of the polyline that share the point $\mathbf{x}_i$. The mathematical definition is as follows. Calling $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$ the points on the polyline immediately preceding and following $\mathbf{x}_i$ respectively, the kink angle is

$$kinkAngle = \cos^{-1}\left(\frac{\langle(\mathbf{x}_{i+1} - \mathbf{x}_i), (\mathbf{x}_i - \mathbf{x}_{i-1})\rangle}{\|\mathbf{x}_{i+1} - \mathbf{x}_i\|\|\mathbf{x}_i - \mathbf{x}_{i-1}\|}\right). \tag{3.1}$$

Of all the kink angles of a branch, the *maximum kink angle* is considered. Inspired by the work of Li *et al.* [96], the following angles are also defined:

- the planar angle between the direction of the y axis $\vec{j}$ and the first segment of the polyline of a branch;
- indicated with $\vec{\mathbf{d}}$ the vector from the starting point of the polyline to the end of it, the planar angle between $\vec{j}$ and $\vec{\mathbf{d}}$;
- the planar angle between a branch and its parent.

The angles between $\vec{j}$ and the first segment of the polyline or the vector $\vec{\mathbf{d}}$ give, respectively, informations about the first local growth direction of a branch and about a global growth direction. Due to the phototropism of plants, the angle between $\vec{j}$ and the first segment of the polyline is expected to be small. All the angles are measured in radians.

There are two other geometrical quantities that can be defined with the angles between adjacent segments of a polyline approximation of a branch. The first is the *mean curvature* of the cane. I adopt the same circle approximation of curvature found in Belyaev [17]. With reference to figure 3.4, let

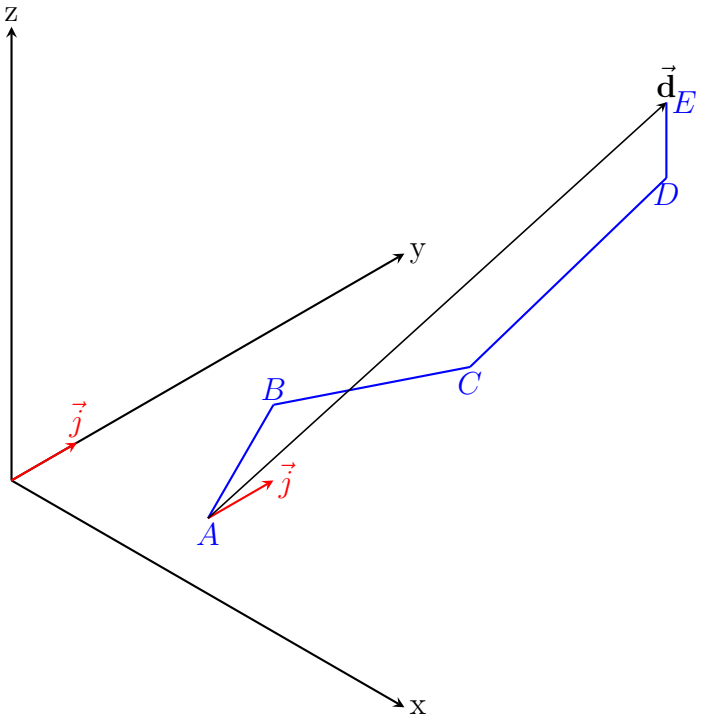


Figure 3.3: Explanation of the angles defined. Suppose the blue polyline is the centreline of a cane. Then, $\cos^{-1}\left(\frac{\langle\vec{j},A\vec{B}\rangle}{\|A\vec{B}\|}\right)$ is the angle between $\vec{j}$ and $A\vec{B}$ and $\cos^{-1}\left(\frac{\langle\vec{j},\vec{\mathbf{d}}\rangle}{\|\vec{\mathbf{d}}\|}\right)$ is the angle between $\vec{j}$ and $\vec{\mathbf{d}}$.

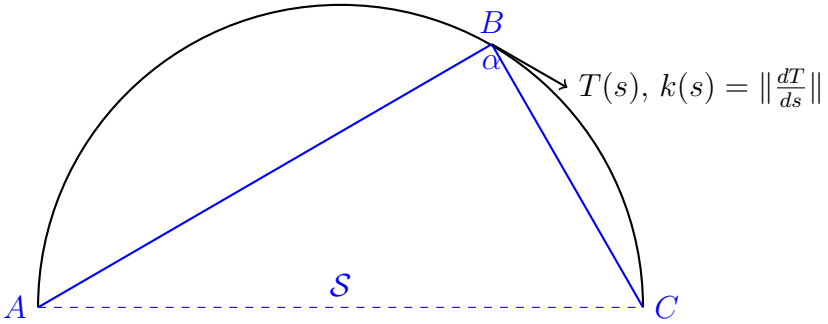


Figure 3.4: Curvature approximation.

the circle be the actual curve locally describing the centreline of a branch and the segments AB and BC its local polyline approximation. Then, the

discrete curvature $\tilde{k}_B$ at B is given by

$$\tilde{k}_B = \frac{4\mathcal{S}}{\|AB\|\|BC\|\|AC\|} = \frac{2\|AB\|\|BC\|\,|\sin\alpha|}{\|AB\|\|BC\|\|AC\|}. \tag{3.2}$$

The discrete curvature, in order to be defined, requires that the polyline approximation consists at least of two segments. Therefore, if a branch is so short that it is just represented with one segment, it will have curvature 0. From the geometry of curves, it is known that the curvature $k(s)$ of the curve, if it exists, is a pointwise defined positive function. However, with the polyline approximation, it can be defined only in certain points, being non zero only at the extremal points of the segments of the polyline. Therefore, it is better defining the mean curvature of the branch as

$$\tilde{k} = \frac{1}{n-1}\sum_{i=2}^{n}\tilde{k}_i, \tag{3.3}$$

where n is the number of segments of the polyline and $\tilde{k}_i$ is the local curvature at the $i-$th point of the polyline. I stress the fact that $\tilde{k}$ is always non negative and that the local curvature is not defined at the starting and ending points of a branch, thus the explanation of the indices in (3.3). I also report that a bound on the error committed in the polyline approximation can be obtained through formula (3.2). For this and an asymptotic discussion of the discrete curvature approximation, I refer again to Belyaev's work [17]. The approximation (3.2) depends, in general, on the discretization of the centreline curve. However, the discrete curvature approximation is often well suited for vine branches, because they usually have "smooth" changes in their growth direction and the linear approximation is usually well fitted. Indeed, for most of the branches, each segment between the buds of a branch is almost straight.

Another quantity that can be defined through angles is a measure of the *bumpiness* of the branch. The mean curvature alone does not give any information about a general "behaviour" of the curve or polyline representation of a branch. Before going through the definition of the bumpiness, I need to introduce a local reference system for branches, that will also be used in the definition of some topological variables. For each branch the vector $\vec{\mathbf{d}}$ is

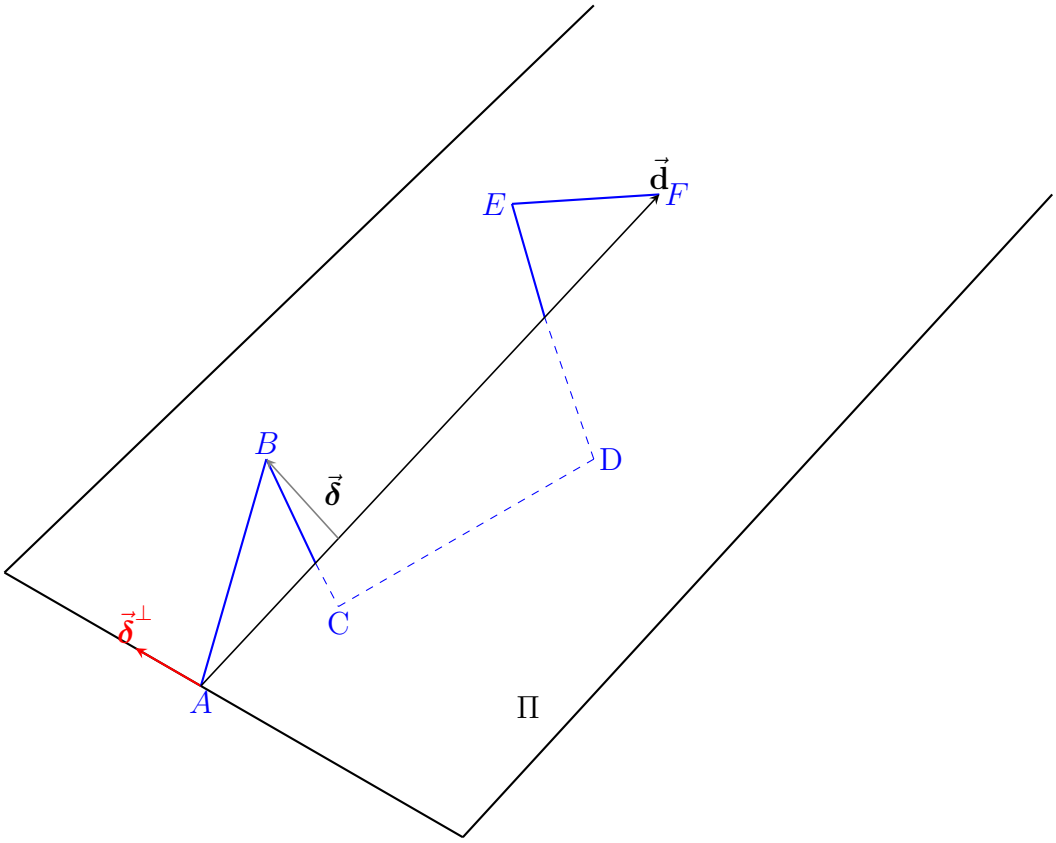


Figure 3.5: Construction of the local reference system $\left\{\vec{\mathbf{d}}, \vec{\boldsymbol{\delta}}, \vec{\boldsymbol{\delta}}^{\perp}\right\}$ for each branch.

defined as the vector from the starting to the ending point of the polyline approximation of the centreline. Then, the vector $\vec{\boldsymbol{\delta}}$ is defined as the vector perpendicular to $\vec{\mathbf{d}}$, connecting $\vec{\mathbf{d}}$ itself to the farthest control point. Finally, the vector $\vec{\boldsymbol{\delta}}^{\perp} = \vec{\mathbf{d}} \times \vec{\boldsymbol{\delta}}$ and the planes $\Pi = \text{span}\{\vec{\mathbf{d}}, \vec{\boldsymbol{\delta}}^{\perp}\}$ and $\Pi' = \text{span}\{\vec{\mathbf{d}}, \vec{\boldsymbol{\delta}}\}$. The start and ending point of a polyline belong to Π and therefore are not considered in the choice of $\vec{\boldsymbol{\delta}}$. The idea behind the bumpiness of a branch is to give a first global information about its shape, while moving along $\vec{\mathbf{d}}$, the main growth direction. For example, if the polyline keeps zigzagging with respect to Π, it can be considered "bumpier" than another branch that is shaped more like a smooth arc. I now proceed to formally define the bumpiness. Given three consecutive control points $\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}$ of a polyline, let $\mathbf{x}_M = \frac{\mathbf{x}_{i-1}+\mathbf{x}_{i+1}}{2}$. Let also $d_\Pi(\mathbf{x}_i)$ be the distance between $\mathbf{x}_i$ and Π. Finally,

let the function f be $(i = 2, \ldots, n-1)$

$$f(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1}) = \begin{cases} 0, & \text{if } d_\Pi(\mathbf{x}_i) \geq d_\Pi(\mathbf{x}_M) \\ 1, & \text{otherwise.} \end{cases} \tag{3.4}$$

So, f takes value 0 every time a triangle made by three consecutive points of a polyline is facing mainly "upwards" with respect to Π. Then, the *bumpiness* is defined as

$$bumpiness = \sum_{i=2}^{n-1} (-1)^{f(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})} \frac{\mathbf{x}_{i-1}\hat{\mathbf{x}}_i\mathbf{x}_{i+1}}{2\pi}. \tag{3.5}$$

In (3.5) the numerator is the acute angle at $\mathbf{x}_i$ in radians.

The last geometric variables that are defined come from the principal component decomposition of the branches. Principal Component Analysis (PCA) is a statistical technique used to investigate and explain the variance-covariance structure of a sample from a geometrical point of view. I refer to Johnson and Wichern [77] for a more detailed discussion on the subject. Therefore, the idea is to apply PCA to branches, considered as samples of points. PCA is applied to both the branches and their "derivative", because hidden information could be present in the derivatives of the branches. By derivative of a branch I mean that the normalised differences between two consecutive endpoints of a polyline are considered. As a result, 6 eigenvectors (principal components) and their eigenvalues are obtained. The eigenvectors give information on how the branches develop in space and the corresponding eigenvalues refer to the importance of those vectors.

3.2.2 Topological variables

The topological variables are defined to take into account information about the *shape* of the branches or topological quantities. As it is known, for example from Sharon and Mumford [138] or Kendall [86], the concept of shape has some intrinsic issues. A *shape* can be defined like a diffeomorphism from the unit circle in itself. So, a space of shapes is a space of diffeomorphism, which is an infinite dimensional space. Therefore, if a finite subset of variables is selected to represent shapes, it can happen that objects with

different shape have the same values for the set of variables selected. This problem can be addressed by considering a quotient space with respect to the selected variables and defining two objects to have the same shape if they have the same values for those variables. This is the same procedure that I follow, with some adaptations. The resulting "space of shapes" of the branches will be briefly analysed in the next section.

In defining topological variables, ideas and methods are drawn from recent fields of research, like Topological Data Analysis (TDA) or the geometry of random fields. In particular, many ideas and definitions are taken from Adler and Taylor [1], Arias-Castro *et al.* [4], Banchoff [10], Cohen-Steiner *et al.* [33], [34], [35], Feragen *et al.* [50], Kendall [86], Sharon and Mumford [138], Turner *et al.* [147] and Wang and Marron [153].

The first straightforward variable to be defined is the *connectivity* of the structure. For each branch, the presence of its parent is recorded by this variable. The connectivity, for how it is defined, is strongly dependent on the quality of the 2D structure extraction set. For this reason, it could be slightly modified. The connectivity could used as a constraint or its definition can be relaxed. The connectivity can be used as a constraint because, given admissible reconstructions, the structure showing more connected parts could be preferred. The second approach consists in relaxing the notion of connectivity to consider particular distributions of the distances between branches. Indeed, not always the parent of a branch is known, but information about connectedness can be inferred if the distance between two branches is known.

Suppose two branches are reconstructed, like in Figure 3.6, the reconstruction is not connected, but a join between them could be likely. Suppose the distances D_i between the starting point of the polyline of a branch to the other $n-1$ branches are given. It is natural assuming that a join can happen between the closest branches. For example with reference to Figure 3.6, $\mathcal{B}_2$ can be joined to $\mathcal{B}_1$ if D is the minimum distance between the starting point of $\mathcal{B}_2$ and any other branch. Now, given the knowledge of some physical characteristics of the machine and of the problem, an admissible-join region can be defined. Because vines are creeper and they grow climbing the wires passing through the posts, joinings mainly happen in a volume that is individuated by the height of the posts multiplied by their width and

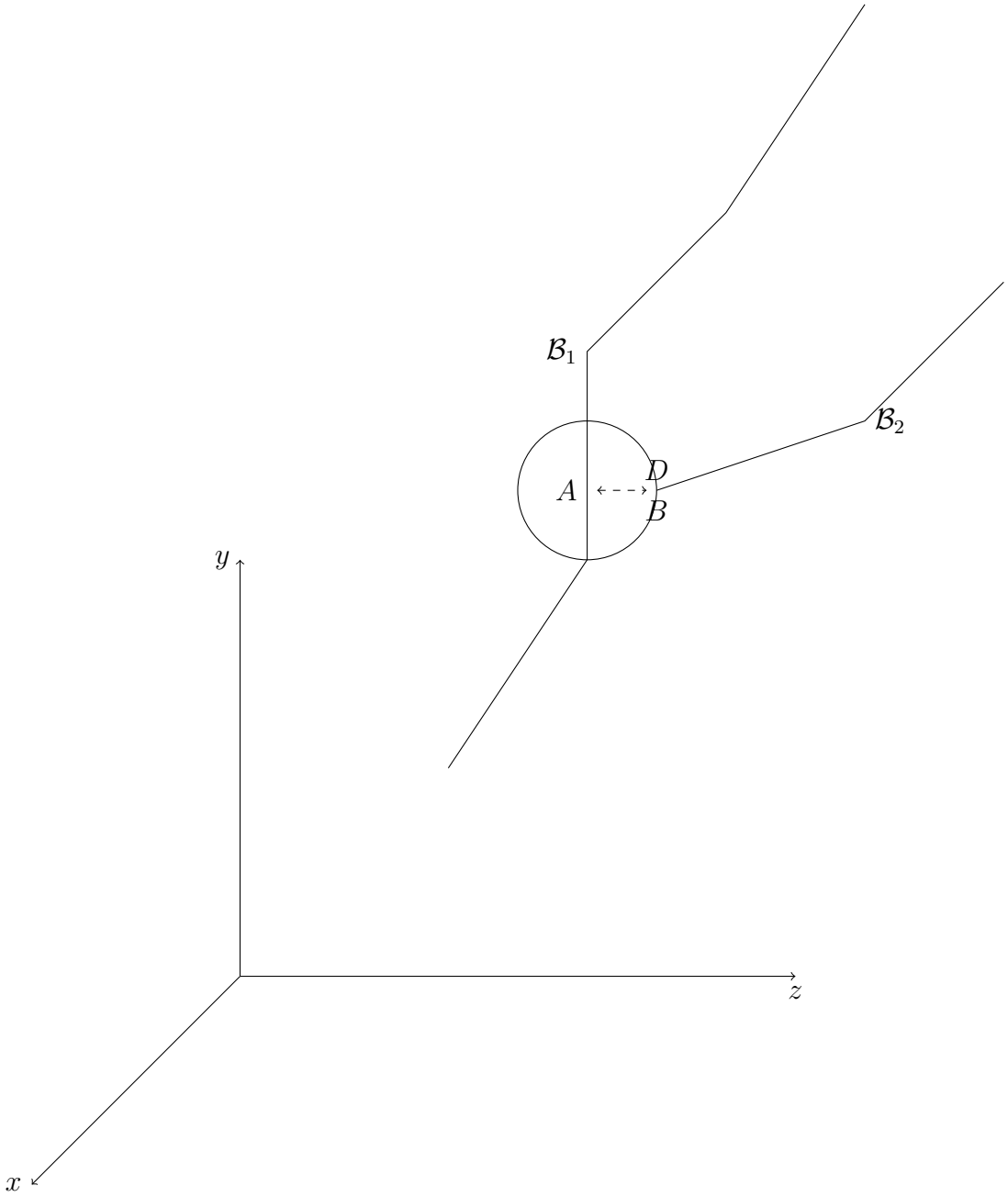


Figure 3.6: Inferring the connectedness of a reconstruction. Suppose the branches $\mathcal{B}_1$ and $\mathcal{B}_2$ are reconstructed as in this figure. The reconstruction is not connected. However, an analysis of the distance D between the starting point of $\mathcal{B}_2$ to $\mathcal{B}_1$ can give information about the connectedness of the structure.

the distance between the posts. Indicated with h the height of the posts, generally of cylindrical shape, with r their width and with L the distance between two consecutive posts, the reconstructions and joinings of branches, must happen in a region whose volume V is $V = h \cdot r \cdot L$. Among these dimensions, the smallest one is r. Therefore, an admissible-join region can be a sphere of radius $\frac{r}{2}$ around the closest point on another branch to the starting point of a considered cane. For example, in Figure 3.6, $\mathcal{B}_2$ can be joined to $\mathcal{B}_1$ if the points A and B have a depth which assumes an admissible value and the distance D is sufficiently small. From this a model on the error committed when reconstructing canes can be proposed to infer about the connectedness of the structure. For example, it can be assumed that, if A is B's closest point on another branch, $B \sim \mathcal{N}(A, \Sigma)$, with $\Sigma = \sigma^2 I_{3\times 3}$, where $\sigma = \frac{r}{2}$ and $I_{3\times 3}$ is the identity matrix in $\mathbb{R}^{3\times 3}$. Then, a distribution on the distance between A and B is easily obtained, since it is well known, for example Johnson and Wichern [77], that $(A-B)^T\Sigma^{-1}(A-B) \sim \chi^2$ with three degrees of freedom. I remark that this model is true *only for the minimum of the distances* between the starting point of a cane and all the other branches and that the distribution *does not depend on the points involved*.

From the previous discussion, a test to check the *global* connectedness of a reconstructed structure can be designed. That is: I want to test

$$H_0: \text{ the reconstructed structure is connected}$$

versus

$$H_1: \text{ the reconstructed structure is not connected.}$$

I indicate with $F(x)$ the cdf of a χ^2 distribution. If n branches are reconstructed, there are $n-1$ i.i.d. minimum distances considered. The cdf of the minimum of those $n-1$ distances can be derived as follows. Let $D = \min\{D_1, \ldots, D_{n-1}\}$. Then

$$\begin{aligned} P(D \leq d) &= P(\min\{D_1, \ldots, D_{n-1}\} \leq d) \\ &= P(\text{at least one } D_i \text{ is smaller than } d) \\ &= 1 - (1 - F(d))^{n-1}. \end{aligned} \tag{3.6}$$

Now, if the probability of observing the overall minimum d_{min} of the distances is small, the structure can be considered disconnected. Therefore, if

$$P(D \leq d_{min}) = (1 - F(d_{min}))^{n-1} \leq 0.05, \tag{3.7}$$

H_0 is rejected at a 95% confidence. This is analogous to computing the $p-$value for a statistic. This will be used in the application of RANSAC algorithm to my problem.

Defining other topological variables is more complicated. Turner *et al.* [147] prove that the Persistent Homology Transform (PHT) is a sufficient statistics for shapes and surfaces, when 2D manifolds are embedded in a 3D space. The PHT is a map that describes changes in the topology of a surface along a given direction. There is a variation of the PHT that relates to the Euler Characteristic χ of a surface. The hypotheses under which χ can be used instead of PHT, hold for the mathematical model of the vines. The definitions and derivation are rather technical, therefore I refer again to the works of Turner *et al.* and Cohen-Steiner *et al.* [35] for further details. However, there are two problems with the Euler Characteristic. The first is related to the mathematical model of branches. To see this, I briefly report a theorem in Banchoff's work [10]. Suppose M is a closed smooth surface without boundaries embedded in $\mathbb{R}^3$ and consider a linear function ξ on $\mathbb{R}^3$ given by projecting all of $\mathbb{R}^3$ to the line determined by a vector $\boldsymbol{\xi}$. A point p of M is said to be a *critical point* for ξ if the tangent plane to M at p is perpendicular to $\boldsymbol{\xi}$; all other points of M are called *ordinary points* for ξ.

Theorem 3.1. *If ξ has a finite number of critical points on M*

$$\sum_{p \text{ critical for } \xi} i(p, \xi) = \chi(M), \tag{3.8}$$

where

$$i(p, \xi) = \begin{cases} 1 & \text{if } p \text{ is a local maximum or minimum} \\ -1 & \text{if } p \text{ is a non-degenerate saddle point.} \end{cases}$$

In the mathematical modelling of a branch, however, the manifolds representing them have a boundary (the edges at extremities of the branches) and in the polyline approximation they are not smooth. Therefore, theorem 3.1

cannot be straightforwardly applied. A first approach to apply theorem 3.1 could be to smooth the polyline representations. Even in this case, however, the manifold obtained has still a boundary. To remove the boundary, a hemisphere with radius equal to the average thickness can be glued at the two extremities of the manifolds. This operation usually introduces a new local maximum and a new local minimum, unless $\vec{\mathbf{d}}$ is parallel to either the first or last segments of the polyline or both. Therefore, (3.8) should be re-written accordingly:

$$\begin{aligned} \sum_{p \text{ critical for } \xi} i(p,\xi) &- I(\|\vec{\mathbf{d}} \times (\mathbf{x}_1 - \mathbf{x}_0)\| > 0) \\ &- I(\|\vec{\mathbf{d}} \times (\mathbf{x}_n - \mathbf{x}_{n-1})\| > 0) = \chi. \end{aligned} \tag{3.9}$$

In (3.9), $I(\cdot)$ is the indicator function of its argument and $\mathbf{x}_i$'s are the extremal control points. Again, there is a problem. The smoothed manifold obtained with the previous operations is topologically equivalent to a sphere, which has Euler characteristic equal to 2. Since the Euler Characteristic is a topological invariant, it will always be 2 for any branch modified with the previous procedure, unless at least one of the two indicator functions in (3.9) is non-zero. This means that, for most branches, the left-hand side in (3.9) is identically equal to zero, independently from the group the branch comes from. Another way to see why the Euler Characteristic does not convey much information is that for most branches, for every maximum or minimum there is always a saddle (in any given direction). The only branches not subject to this are precisely the ones having $I(\|\vec{d} \times (\mathbf{x}_1 - \mathbf{x}_0)\| > 0) = I(\|\vec{d} \times (\mathbf{x}_n - \mathbf{x}_{n-1})\| > 0) = 0$. However, the event of finding a branch with at least one of the indicator functions in (3.9) equal to zero, has low probability. Therefore, the relevant topological information, for the branches, are expected to be contained mainly in the maxima and minima along a given direction. So, instead of the Euler Characteristic, the number of maxima and minima of the branches (as manifolds) are recorded, which are variables more closely related to the PHT itself. Another possible solution to this problem is suggested again in Banchoff [10]. If an approximation with polyhedra of the branches is considered, then a discrete version

of theorem 3.1 still holds. An equation like (3.8) is still valid and $\vec{\boldsymbol{\delta}}$ is the direction to choose, because it can be proved to be *general* in the sense of Banchoff [10]. Again, I refer to his work for further details.

Other possible topological variables take into account some particular configurations of the control points, with respect to the local reference system $\{\vec{\mathbf{d}}, \vec{\boldsymbol{\delta}}, \vec{\boldsymbol{\delta}}^{\perp}\}$. Ideas in this section are inspired from Adler and Taylor [1], Arias-Castro *et al.* [4], Kendall [86], Sharon and Mumford [138], Schneider and Weil [136] and Wang and Marron [153]. Conventionally, I consider the plane Π and Π' positively oriented in the directions of $\vec{\boldsymbol{\delta}}$ and $\vec{\boldsymbol{\delta}}^{\perp}$, respectively. Then, the following properties are defined:

1. *a control point has the property A* if it is above Π (in the sense of $\vec{\boldsymbol{\delta}}$) and the *property B* if it is below Π;

2. *a control point has the property L* if it is above Π' (in the sense of $\vec{\boldsymbol{\delta}}^{\perp}$) and the *property R* if it is below Π'.

For each control point, the previous properties are recorded. In the following section I explain the analysis conducted with those properties and how they define a "space of shapes" for the branches. Properties L and A are considered independent, but A and B can have a dependency, as well as L and R. $|A|$ and $|L|$ are the number of control points having the properties A and L respectively, for each given branch. Clearly, the starting and ending control points do not have any of those properties. So, if there are n control points in a branch, only $n-2$ can have the properties A or L. The proportions $p_A = \frac{|A|}{n-2}$ and $p_L = \frac{|L|}{n-2}$ are also recorded. Other quantities defined are: *the number I of intersections between the polyline and* Π and *the number T of intersections between the polyline and the plane* Π'. The previous variables record some aspects of the shape of the polylines and analysing them could give information about how to match branches.

3.2.3 Other variables

With the term other variables, quantities that do not belong to any of the previous classes are designated. The pruning machine has three cameras, but

only two of them are used for branch correspondences, depending on the specific main direction of the considered branch. If a cane is mainly horizontal, the correspondences are sought using the Top and Right image, otherwise the Left and Right image are used. This procedure helps in reducing ambiguity. Moreover, the *back-projection error* in the image not used for the correspondences can be measured. Similar to the back-projection error, the variable measuring the *overlapping* can be defined. This quantity measures how well a back-projected reconstructed branch covers the 2D canes that originated it. Finally, the *endpoints agreement* is defined and it is related to the connectedness. This variables checks the agreement of the endpoints of matching branches. For example, if a branch in one image is forking out another cane, then the matching candidate has to fork out of another branch as well.

3.3 Space of shapes of branches

The idea behind the previous definition of the properties A, B, L and R is to have a new parametrization of the branches that allows for a comparison of their centrelines, but *considering only their shape*. The original expectation is that a reconstruction given by a wrong correspondence would generate an irregular branch, with this irregularity reflecting in the particular shape of the branch. Therefore it is reasonable to think that the shapes of the two groups would be different. Many ideas in this section are inspired from Wang and Marron [153] and Sharon and Mumford [138]. I now discuss how those variables can define a suitable space of shapes, that can be made into a vectorial space, up to an equivalence relation. Consider a branch and its local reference system. To eliminate any interference due to the geometry, suppose that the endpoints of the segments of the polylines belong to a discrete grid, like in Figure 3.7 and around the intersections of the planes Π and Π', like in figure 3.8. For every point a new representation is defined, as in Figure 3.8. A new pair $(\alpha, \beta)_i$ is associated to every point $\mathbf{x}_i$ with the following rules:

- $\alpha = 1$ if $\mathbf{x}_i$ is above Π in the sense of $\vec{\delta}$, $\alpha = 0$ if $\mathbf{x}_i$ belongs to Π and $\alpha = -1$ otherwise (i.e.: $\alpha = 1$ if $\mathbf{x}_i$ has property A, $\alpha = -1$ if $\mathbf{x}_i$ has property B and $\alpha = 0$ otherwise);

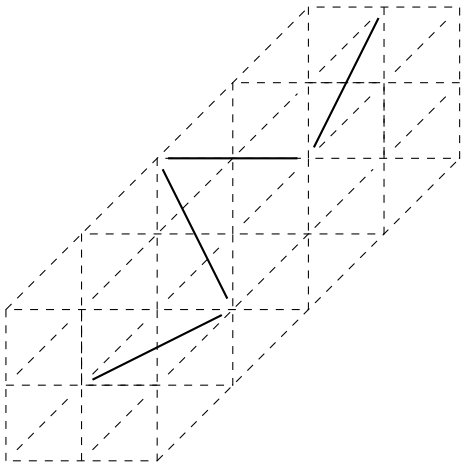


Figure 3.7: Discrete grid idealization of a branch.

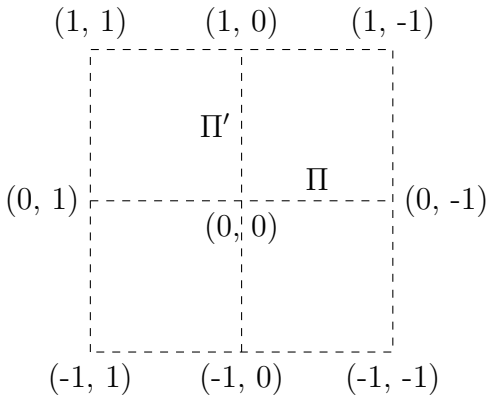


Figure 3.8: Front view of the coordinates of each control point in the shape representation of the branches.

- $\beta = 1$ if $\mathbf{x}_i$ is above Π' in the sense of $\vec{\delta}^{\perp}$, $\beta = 0$ if $\mathbf{x}_i$ belongs to Π' and $\beta = -1$ otherwise (i.e.: $\beta = 1$ if $\mathbf{x}_i$ has property L, $\beta = -1$ if $\mathbf{x}_i$ has property R and $\beta = 0$ otherwise).

Now, the geometrical representation $\{\mathbf{x}_0, \ldots, \mathbf{x}_n\}$ of a branch is replaced by the new *shape* representation $\{(0,0), (\alpha,\beta)_2, \ldots, (\alpha,\beta)_{n-1}, (0,0)\}$.

Example. I illustrate with an example how a branch is mapped into the new representation. With reference to Figure 3.9, suppose a polyline

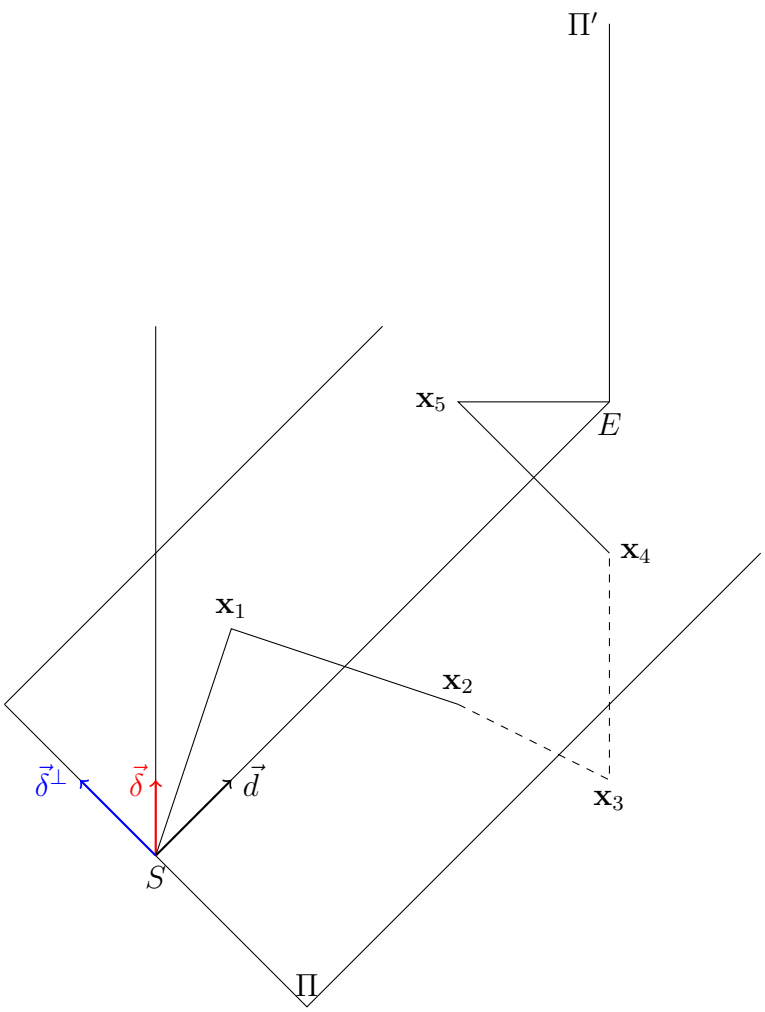


Figure 3.9: Example used to illustrate the shape representation

$\{S, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, E\}$ is given.

1. $\mathbf{x}_1$ belongs to Π' and is above Π;
2. $\mathbf{x}_2$ belongs to Π and is to the right of Π';
3. $\mathbf{x}_3$ is below Π and to the right of Π';
4. $\mathbf{x}_4$ belongs to Π and is to the right of Π';
5. $\mathbf{x}_5$ is above Π and is to the left of Π';

Then, following the rules given before,

1. $\mathbf{x}_1$ is mapped to $(1, 0)$;

2. $\mathbf{x}_2$ is mapped to $(0, -1)$;

3. $\mathbf{x}_3$ is mapped to $(-1, -1)$;

4. $\mathbf{x}_4$ is mapped to $(0, -1)$;

5. $\mathbf{x}_5$ is mapped to $(1, 1)$;

So that the new shape representation for the polyline in the example is $\{(0, 0), (1, 0), (0, -1), (-1, -1), (0, -1), (1, 1), (0, 0)\}$.

Mumford and Sharon in [138] defined a shape, for a closed planar curve, to be the image of a diffeomorphism from the unit circle to itself. In the case of a polyline, its shape can then be defined extending the previous definition. Therefore, the shape of a branch can be considered as the image of a diffeomorphism from a unitary 3D segment to itself. Now, the following facts are straightforward:

1. the new representation describes the shape of the centreline of a branch in the sense that it is necessary and sufficient to identify the points and therefore a diffeomorphism a.e. (for example with piecewise affine linear functions) from a unitary 3D segment to itself having a given branch as its image;

2. the new representation is independent from rotation, translation or dilatation (of the type $\{c\mathbf{x}_0, \ldots, c\mathbf{x}_n\}$) of $\{\mathbf{x}_0, \ldots, \mathbf{x}_n\}$, because the properties A, B, L and R do not change (if Π and Π' are rotated or translated accordingly);

3. the relation $\{\mathbf{x}_0, \ldots, \mathbf{x}_n\} \sim_T \{\mathbf{y}_0, \ldots, \mathbf{y}_n\}$ if their shape representations are the same is an equivalence relation .

Another more intuitive explanation of why the new representation is related to the shape of branches is the following. Each branch could be represented and identified by specifying:

+	-1	0	1
-1	-1	-1	0
0	-1	0	1
1	0	1	1

Table 3.1: Addition table for two shapes

1. the length of each segment of the polyline;

2. two unsigned angles (for every pair of segments of the polyline) in the planes Π and Π';

3. the representation (α, β) for each control point.

The first two quantities refer to geometrical entities. If those are neglected, the remaining attributes must refer to something different from the geometry. In this case it is a primitive concept of shape. Now, if the quotient space $\{\text{shape representations}\} / \sim_T$ is considered, a *space of shapes* for branches is obtained. This gives to the branches a particular "fingerprint", similar to the work of Sharon and Mumford [138]. However, there is an important remark that needs to be made precise: the space is defined for each fixed number $n \geq 3$ of endpoints of the polylines.

To do statistical analysis in these new spaces, notions of addition and multiplication by a constant need to be defined; that is, the spaces need to be given a vectorial space structure. With an abuse of notation I write an equivalence class in the space of shapes as a shape representation. Multiplication by a constant c is defined in the following way:

$$c \cdot \{(0,0), (\alpha,\beta)_1, \ldots, (\alpha,\beta)_{n-1}, (0,0)\} =$$

$$= \begin{cases} \{(0,0), (\alpha,\beta)_1, \ldots, (\alpha,\beta)_{n-1}, (0,0)\} & \text{if } c > 0 \\ \{(0,0), (0,0), \ldots, (0,0)\} & \text{if } c = 0 \\ \{(0,0), (-\alpha,-\beta)_1, \ldots, (-\alpha,-\beta)_{n-1}, (0,0)\} & \text{if } c < 0. \end{cases}$$

Addition between two shapes, with given n, $\{(0,0), (\alpha,\beta)_1, \ldots, (\alpha,\beta)_{n-1}, (0,0)\}$

and $\{(0,0),(\gamma,\delta)_1,\ldots,(\gamma,\delta)_{n-1},(0,0)\}$ is defined as the new shape $\{(0,0),(\alpha+\gamma,\beta+\delta)_1,\ldots,(\alpha+\gamma,\beta+\delta)_{n-1},(0,0)\}$, where each component of the pairs is computed according to the addition Table 3.1.

Having defined an addition between shapes and a multiplication by a constant, a notion of *mean shape* can be defined. Defining a variance in this space is more complicated; I will consider a simpler *dispersion measure*, given by:

$$\forall i = 1,\ldots,n-1,\ \left\{(0,0),\frac{1}{m}\sum_{j=1}^{m}\left(|\alpha_j^i-\bar{\alpha}_j^i|,|\beta_j^i-\bar{\beta}_j^i|\right),(0,0)\right\}, \qquad (3.10)$$

where $\bar{\alpha}_j^i$ is the α_j^i of the mean shape of the group and analogously for $\bar{\beta}_j$ for each j. m is the number of branches with a given n and the division is meant in the usual sense in $\mathbb{R}$.

As I outlined earlier, the topology of branches is analysed in the hope of finding a clear distinction between correct and wrong correspondences. However, the analyses did not give the results I expected. There is no sensible difference between the distributions of the α's and β's for any given n. Moreover, it is suspected that, when some small differences are observable, they are due to randomness, rather than to belonging to one specific group. This deduction follows from the belief that if a difference was due to the groups, then it would be observed more often. Also, the graph of the dispersions of the two groups does not suggest the presence of a significant specific shape in any of the two groups. The suspect is that the optimiser used in the reconstruction process tries to give a realistic appearance to the branches even when wrong matches generate possibly irregular curves. As an example of analysis, I only show the α_i's for the polylines with 15 endpoints. The analyses were conducted separately on α's and β's. This is in accordance with the hypothesis of independence between properties A and L. This also makes the visualisation simpler, because it means the curves representing the shapes of the branches can be projected on lower-dimensional (1D) spaces and therefore represented as graphs of piecewise linear functions. In Figure 3.10 the α's of the branches belonging to the group of correct matches can be seen. Figure 3.11 shows the α's of the branches belonging to the group of

wrong matches. There are no clear predominant shapes in the two groups, they seem uniformly randomly distributed. In Figure 3.12, the mean α's of the two groups are shown. While it seems the mean shapes are different, Figures 3.13 and 3.14, that present the dispersions of the α's, suggest that any difference may be due to randomness and not to the belonging of a group, because the variability, at any point, is high. I finally acknowledge here, that, even though this analysis did not revealed entirely useful in this case, because of the probable bias due to the reconstruction algorithm, it can be very effective in other fields. Moreover, besides gaining insights in the "shapes" of curves, a way to represent (equivalence classes of) 3D curves in lower dimensional spaces has been obtained.

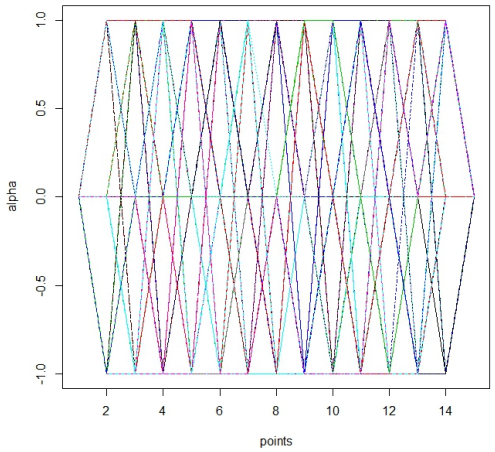


Figure 3.10: Projection of the first component (α) of the curves representing the shape of the ground truth data. The axis $\alpha = 0$ corresponds to the plane Π.

3.4 Statistical learning and analysis of the variables

After having listed the variables used to build model of the vines, I discuss here how the distributions of those variables are learnt. This step is necessary to gain a better knowledge of the structures and to implement the methods used to seek for the correspondences. As seen in Section 3.2, some correspondences imply a reconstruction. Given the reconstruction, the probability of

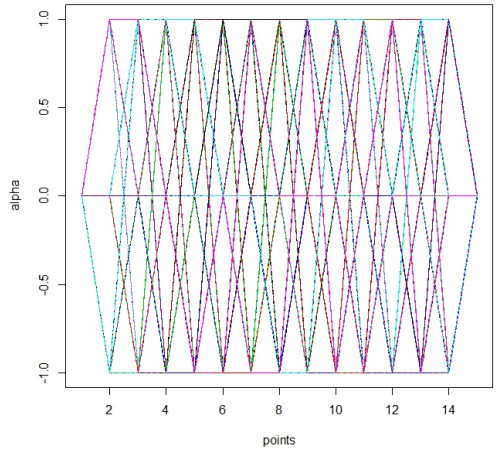


Figure 3.11: Projection of the first component (α) of the curves representing the shape of the wrong correspondences.

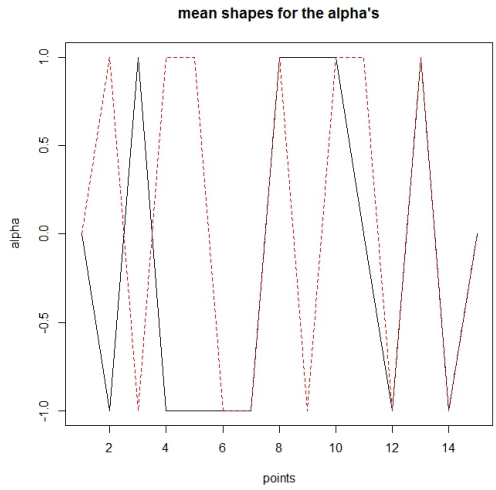


Figure 3.12: α component of the mean shapes of the two groups. The dashed line is the mean shape of the wrong correspondences.

a match being correct can be inferred using some constraints and the previously described variables. Therefore, I need to know the distributions of the variables in case of good reconstructions. At the same time, information about incorrect matchings are gathered. In this way, it can be more accurate

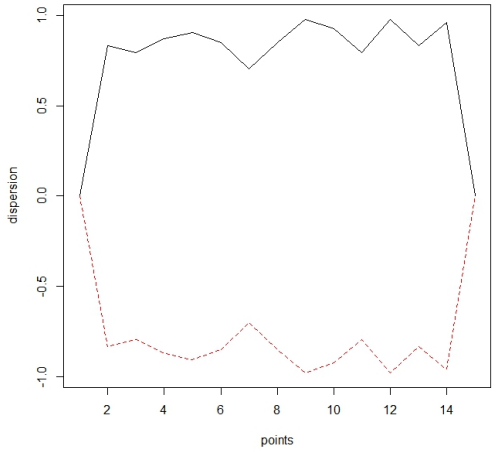


Figure 3.13: Dispersion for the α of the correct correspondences.

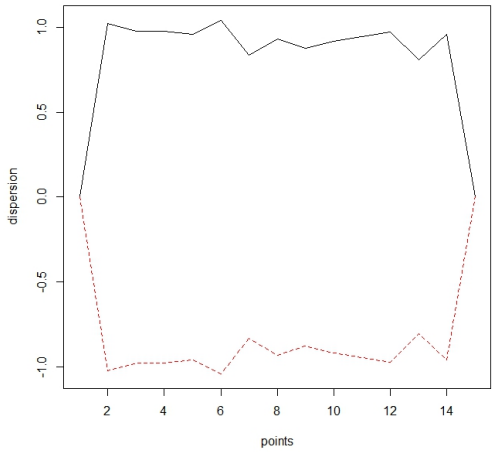


Figure 3.14: Dispersion for the α of wrong correspondences.

to classify and decide between admissible or wrong correspondences.

To learn the distribution of the variables, I need to observe a sample coming from the set of correct correspondences of branches. So, a set of ground truth data is manually selected from images of vines. An example of manually selected labelled canes can be seen in Figure 3.15. The ground

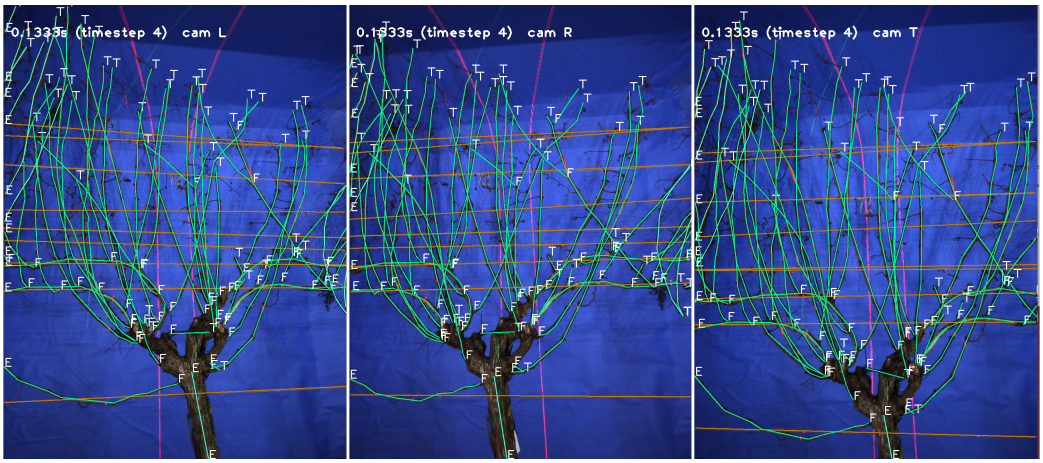


Figure 3.15: Manually labelled ground truth data.

truth labelling of data was performed by another researcher associated with the project. The 3D reconstructions are then computed and the variables measured. At the same time, wrong correspondences are generated and the features observed. A total of 453 ground truth matchings are found and 2943 wrong correspondences computed. The first step into learning the distributions of the two groups is an exploratory analysis of the datasets. Boxplots, scatterplots, and pairwise QQplots are computed.

The boxplots in Figure 3.16, for example, suggest a greater variance in the group of wrong correspondences. However, by looking at the plots in Figure 3.17, some more qualitative observations can be drawn. For example, it appears that the distributions of some of the geometrical variables are different over the two groups of correspondences. Before proceeding into the analysis, the following assumptions are made:

1. the geometrical variables are independent from the topological ones. Geometry and topology of manifolds are related, but I am assuming independence between the two classes of variables I specifically defined.

2. Among the geometrical variables, the triple (curvature, length, thickness) is independent from the others as well as the variables coming from PCAs techniques.

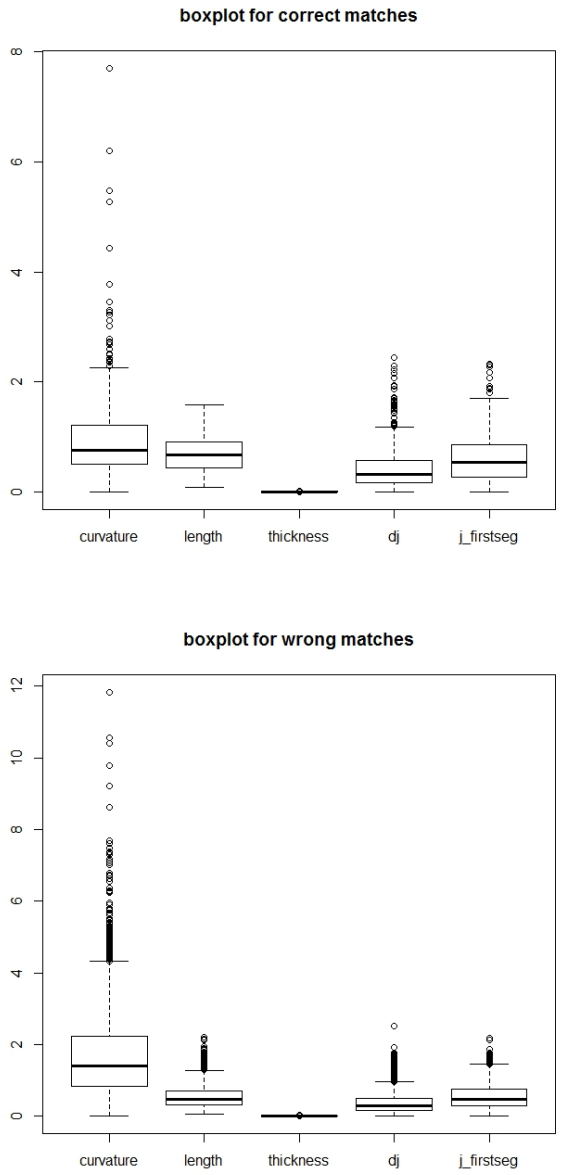


Figure 3.16: Boxplots of some of the geometric variables. The thick black line in the box is the median of the data. The upper and lower edges of the boxes are the first and third quartiles Q_1 and Q_3. The upper and lower extremities of the whiskers are, respectively, $Q3 + 1.5 \cdot IQR$ and $Q1 - 1.5 \cdot IQR$, IQR being the interquartile range: $IQR = Q_3 - Q_1$. The units are: metres^{-1} (curvature), metres (length), cm (thickness) and radiants (angles).

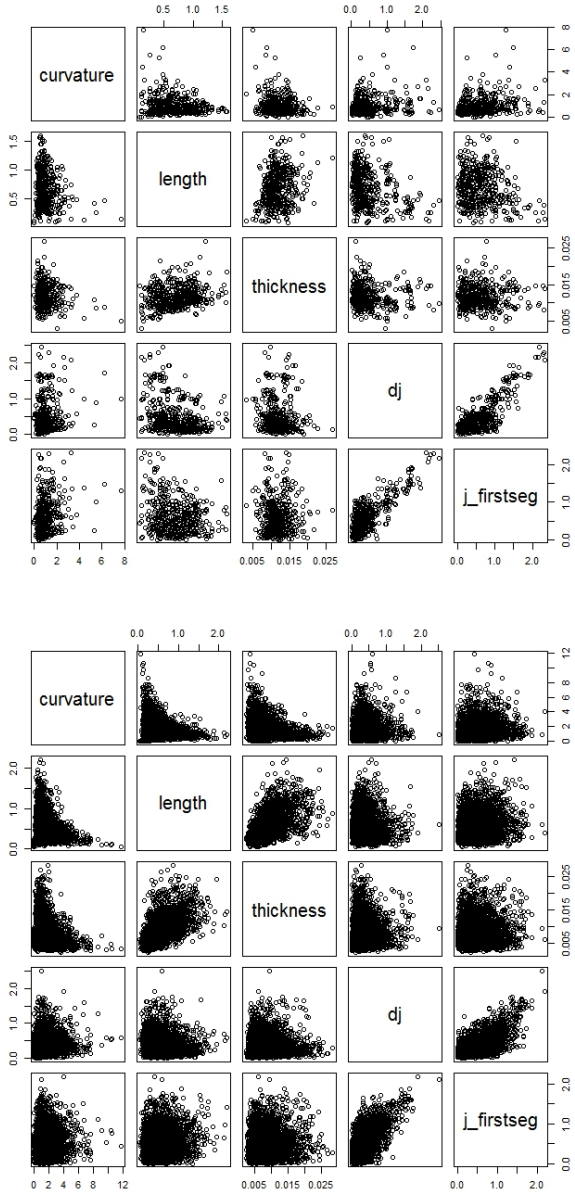


Figure 3.17: Scatterplots of the geometric variables shown above. Top scatterplot: ground truth matchings, bottom scatterplot: wrong ones.

3. The couple (bumpiness, maxKinkAngle) is independent from the other variables measuring angles. This is reasonable, since they deal with different sets of angles.

independence	(curv, len, thick)	PCA var	(bumpy, maxK)	angles var
(curv, len, thick)	N	✓	✓	✓
PCA var	✓	N	✓	✓
(bumpy, maxK)	✓	✓	N	✓
angles var	✓	✓	✓	N

Table 3.2: Table summarising the dependence relationships among the geometrical variables. The ✓ indicates that the group of variables are independent.

The independence relationships are summarised in Table 3.2. These assumptions further divide the set of geometrical variables in subgroups: a group containing curvature, length and thickness, one with bumpiness and maxKinkAngle, one containing the PCA variables and finally the remaining angles. Clearly, some errors are generated with the above assumptions. For example, curvature and bumpiness could be related. However, the triple (curvature, length, thickness) by itself can convey information about the phototropism of the vines and thus in classifying the correct correspondences.

To investigate dependencies linking curvature, length and thickness, regression functions are computed. The analysis of the regression models does not reveal significant differences in the relation linking curvature, length and thickness in the two groups. The converse happens with models linking the angles between $\vec{\mathbf{d}}$ and $\vec{\mathbf{j}}$ with the angles between $\vec{\mathbf{j}}$ and the first segment of the polyline. The model

$$curvature = \beta_0 + \beta_1 length^{-1} + \beta_2 thickness^{-1} + \varepsilon$$

is found to be the best one for both groups, with $R^2 \approx 0.57$ for the class of correct correspondences and $R^2 \approx 0.72$ for the wrong ones. In every case, a 95% confidence interval of corresponding coefficients in the two models overlap. An explanation of these facts could be found in the high variability of the variables of the group of wrong correspondences together with a bias introduced by the reconstructing algorithm. The models linking the angles

are found to be

$$\text{angle}(\vec{d}\vec{j}) = 0.36 + 0.53\ \text{angle}(\vec{j}\text{first segment}) + \varepsilon_I,$$

with $R^2 \approx 0.60$ and $\varepsilon_I \sim \mathcal{N}(0, 0.07)$ for the group of correct correspondences and

$$\text{angle}(\vec{d}\vec{j}) = 0.207 + 0.36\ \text{angle}(\vec{j}\text{first segment})^2 + \varepsilon_O,$$

with $R^2 \approx 0.44$ and $\varepsilon_O \sim \mathcal{N}(0, 0.04)$ for the group of wrong ones. However, when the angles are close to 1 radian, there is not much difference between the two models.

Subsequently, Kolmogorov-Smirnov tests and density kernel estimations are performed. To allow more flexibility, a non-parametric multivariate density estimation is adopted. Some remarks and assumptions are needed for this step. Given the high dimensionality (19 dimensions) of the vector of random variables considered, the density estimation is problematic. However, thanks to the assumptions made above, simple multivariate density estimations can be performed and the marginals computed. This is better than univariate estimations for each single variable. Indeed, univariate density estimation would recover the pdf of a variable *conditionally* to all the others. A multivariate density estimation is performed for the triple (curvature, length, thickness), for the couple (bumpiness, maxKinkAngle) and for the other angle variables. The multivariate densities $\hat{f}_{\mathbf{H}}(\mathbf{x})$ are recovered by

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n} \sum_i K_{\mathbf{H}}(\mathbf{x} - \mathbf{x}_i).$$

The standard multivariate normal kernel is used. The bandwith (or smoothing) matrices $\hat{H}$ for the multivariate density estimations are computed minimising the AMSE (Asymptotic Mean Square Error), Wand and Jones [152]. Figure 3.18 shows the multivariate density estimation of the triple(curvature, length, thickness) for the set of correct correspondences. Given the multivariate densities, the marginal for each variable is computed. As an example, Figure 3.19 shows the marginals of the curvature in the two groups. The densities of the topological variables are estimated independently and in an univariate way. An univariate estimation for the topological variables is le-

gitimate, because they are assumed independent from one another. This may not be necessarily true for the number of maxima and minima. For example, for every two maxima there is a minimum or for every two minima there is a maximum. However, it is a difficult relationship to model, without an analytic expression of the manifolds representing the branches.

Because of the similarities between the distributions of the variables

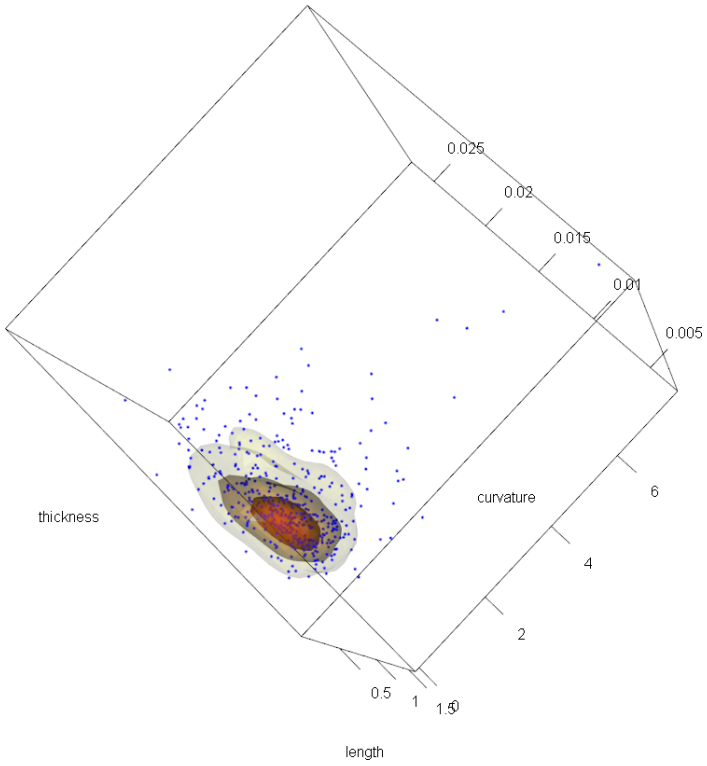


Figure 3.18: Estimated multivariate density for (curvature, length, thickness) for the correct correspondences. Darker zones correspond to higher densities.

in the two groups, I used the Kolmogorov-Smirnov test at a 95% significance level to decide about their equality. To have stronger results, since the number of data points for wrong matches is greater than the sample size for correct matches, a bootstrap approach is used. For each variable, 1000 bootstrap samples of size 453 from the group of wrong matches are generated. Then, a two sided Kolmogorov-Smirnov test is performed individually

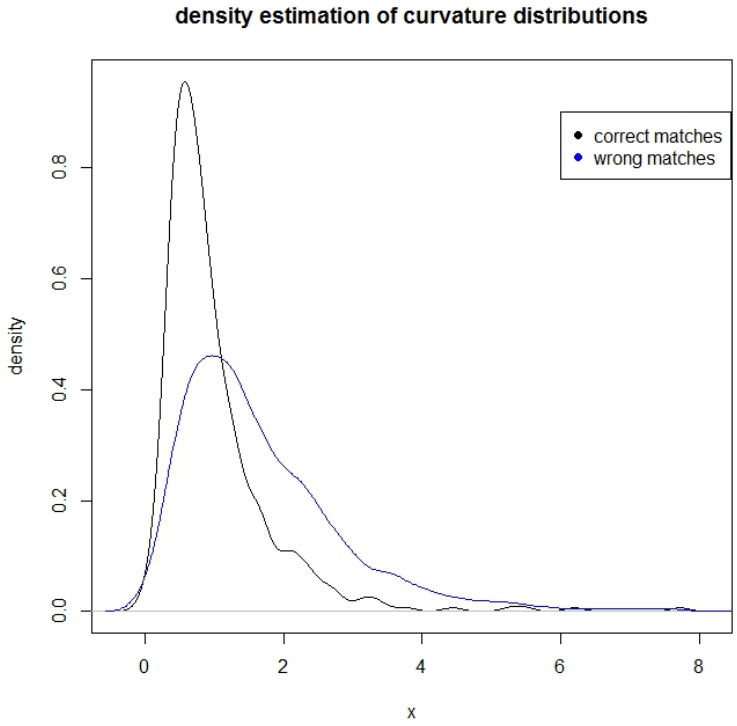


Figure 3.19: Estimated densities of curvature for ground truth data (black) and wrong matches (blue).

for each bootstrap sample against the group of correct matches. The results are recorded and a decision on the equality of the distributions is reached from the bootstrap tests. The tests on curvature, length, thickness and the angles between $\vec{d}$ and $\vec{j}$ and $\vec{j}$ and the first segment of the polylines reject the null hypothesis on the equality of the distributions. The converse is true for other geometrical variables such as bumpiness or angles between branches. In case of the eigenvalues obtained through the PCA of the branches and their derivatives, the distributions seem to be different between the groups; this suggests a distinct geometry in case of correct and wrong matches. This aspect is consequently reflected in the directional analysis of the corresponding eigenvectors. The Kolmogorov-Smirnov test, whose results can be seen in Table 3.3, gives already some informations on the variables that could be more meaningful for computing the correct correspondences. For example

the angles between canes or the bumpiness do not convey much information about the groups. This conclusion is unexpected in the case of bumpiness, as well as the other not satisfactory results regarding topological variables. Bumpiness, though computed from geometrical quantities, is more of a variable describing a global characteristic about the shape of branches. The topological variables were defined with the expectation that reconstructing branches from wrong correspondences would give not regular 3D reconstructions. The data show that this is not the case. Only small differences are noticeable between the correct and wrong set of correspondences if only the topological variables are taken into account. The suspect is that this is probably due to the optimiser used in the reconstruction of the branches that forces them to look more realistic. The topological variable having the greatest importance is the connectedness: the ground truth data generate a connected structure with a probability which is more than 7 times higher than the wrong correspondences. However, it is likely that the high significance of the connectedness is due to having manually selected the correct correspondences. The number of maxima and minima also shows a slightly different behaviour in the two groups. Moreover, the data confirm the choice of considering only the maxima and minima, rather than the Euler characteristic. Only 0.4% and 0.6% of the branches in the set of correct and wrong correspondences, respectively, have *approximately* at least one of the indicator functions in (3.9) equal to zero. All the other topological variables do not look significant.

3.5 Conclusions

Once the distribution of the variables has been investigated, the information provided are used to decide if a new found match is assigned to the set of correct or wrong correspondences. The methods used to decide how a new match is assigned to either of the groups can be divided into two classes, depending on a direct computation of the correspondences or an estimation of the marginal probabilities of the matchings. The first one includes a maximum likelihood method, RANSAC and a tree matching algorithm. The other class includes a decision method based on a Bayes classifier, one on a

KS test	result
curvature	different
length	different
thickness	different
angle($\vec{dj}$)	different
angle($\vec{j}$ first segment)	different
angle parent-child	not different
bumpiness	not different
maxKinkAngle	slightly different
PCA, DerPCA	different
connectedness	different
maxima	slightly different
minima	slightly different
p_A	not different
p_L	not different

Table 3.3: Table showing the conclusions drawn from the Kolmogorov-Smirnov test. With "slightly different" it is meant that the $p-$value is smaller than 0.05, but close to 0.05.

Support Vector Machine model and another one based on a Gradient Boosting method for classification. Table 3.4 gives a summary of all the variables involved and in what methods they will be used.

In general, I am interested in estimating a function

$$F(M(l,r)|\mathcal{I},\mathcal{O},\mathcal{C}), \tag{3.11}$$

that decides the group a new match (l,r) belongs to, given a set $M(l,r)$ of measurements of the 3D reconstruction associated to (l,r), models $\mathcal{I}$ and $\mathcal{O}$ of the correct and wrong correspondences, respectively, and a set $\mathcal{C}$ of constraints. In the next chapter I will focus on the Bayes Classifier method and those based on support vector machine and a gradient boosting method. The generic function F in (3.11), for the methods of the next chapter, will be a probability. In Chapter 5, I will present the remaining approaches based on a suitable modification of state-of-the-art algorithms so that they can be applied to my problem. These methods are: a maximum likelihood, RANSAC

and tree matching. In case of maximum likelihood, the function in (3.11) will be a likelihood and a function checking the consistency of a match given a proposed generated model in RANSAC. Finally, in case of tree matching, (3.11) will be the cost associated to a given set of matchings.

Chapter 3 achieves the third objective in the list of Section 1.4. I investigated possible variables to describe key aspects of branched structures and the relations among them. These features will be used in the following to search for suitable models of the vines in order to find the correct correspondences between the images.

Table 3.4: Summary of the variables used in the following in the different methods. BC = Bayes Classifier, GBM = Gradient Boosting Method, SVM = Support Vector Machine, AVE = Averaged method (GBM, SVM and BC), ML = Maximum Likelihood, GM = tree matching.

Variable	**Brief. Description**	**Type**	**BC**	**GBM**	**SVM**	**AVE**	**RANSAC**	**ML**	**GM**
curvature	curvature of branches	continuous	yes	yes	yes	yes	yes	yes	yes
length	length of branches	continuous	yes	yes	yes	yes	yes	yes	yes
thickness	width of branches	continuous	yes	yes	yes	yes	yes	yes	yes
angle($\vec{dj}$)	angle in $[0, \pi]$	continuous	yes	yes	yes	yes	yes	yes	yes
angle($\vec{j}$ first segment)	angle in $[0, \pi]$	continuous	yes	yes	yes	yes	yes	yes	yes
angle parent-child	angle in $[0, \pi]$	continuous	no	yes	no	yes	no	no	no
bumpiness	"shape" parameter	continuous	no	yes	no	yes	no	no	no
maxKinkAngle	angle in $[0, \pi]$	continuous	yes	yes	yes	yes	yes	yes	yes
PCA, DerPCA	"eigenvalues" of branches	continuous	no	yes	no	yes	no	no	no
connectedness	connectedness of structure	discrete	yes	yes	yes	yes	yes	yes	yes
maxima	#maxima (as a manifold)	discrete	yes	yes	yes	yes	no	yes	no
minima	#minima (as a manifold)	discrete	yes	yes	yes	yes	no	yes	no
p_A	prop. of "A" points	discrete	no	yes	no	yes	no	no	no
p_L	prop. of "L" points	discrete	no	yes	no	yes	no	no	no
endpoint agreement	type of endpoints	discrete	yes	yes	yes	yes	no	yes	no
overlapping	% overlap in back-proj. canes	continuous	yes	yes	yes	yes	yes	yes	no
back-proj. error	displacement in back-proj. canes	continuous	yes	yes	yes	yes	yes	yes	no

Chapter IV

Proposed methods for computing probabilities of matches

4.1 Introduction

In the present chapter I describe a first class of methods used to address the correspondence problem in the case of branched structures. This chapter constitutes one of my biggest contributions. The algorithms here discussed build a model, exploiting possible relations between variables, to identify and discriminate correct and wrong correspondences. Subsequently, they compute the probability of a match being correct, conditionally on the other matches, based on the fitted model. Therefore, one of the main aim is to find good methods to compute probabilities that matches are correct, conditionally on other matches. In other terms, the common idea behind this class of methods is the following. Given some ground truth data, a statistical classifier is built. Then the decision function of the classifier is used to compute the probability of a new given match being correct.

Three well known classifiers are considered for my problem: a Bayes classifier, Support Vector Machine and a Logit classifier boosted through a functional gradient descent algorithm. Finally another algorithm is proposed: it is an averaging of the algorithms here presented.

In the next section of this chapter I describe a general approach to compute the probabilities of correspondences being correct. Then, I explain an algorithm based on Gibbs sampling to select the correspondences judged as correct. This part of the exposition is a modification of the work by Botterill *et al.* [20], which was the initial method used in this project to find the correct correspondences. It was also one of the starting points of my research. Subsequently, I discuss the construction of the Support Vector Machine clas-

sifier. Section 4.4 explains the boosting of the Logit algorithm through the functional gradient descent and the last section describes the averaging of the methods here discussed.

4.2 A general framework to find matches

The following discussion reflects the work by Botterill *et al.* [20], with some modifications in order to accommodate some changes I brought in the method. Specifically, the modifications relate to the structure of the set Ω defined below and how the likelihoods for candidate matchings are defined.

Let L and R be the left and right images and, recalling the convention in Chapter 3, let $|L|$ and $|R|$ the number of branches seen in L and R, respectively. If $|L| \neq |R|$ there will surely be some branches that cannot be matched. Therefore, at most $m = \min\{|L|, |R|\}$ matches can take place and let $M = \max\{|L|, |R|\}$. I assume I can refer to the branches in the images with unique indices $i = 1, \ldots, |L|$ and $j = 1, \ldots, |R|$. $\omega_{ij} = (l_i, r_j)$ indicates the matching between two generic branches l_i and r_j. In case a branch is not matched, I indicate it with $\omega_{i0} = (l_i, \emptyset)$ or $\omega_{0j} = (\emptyset, r_j)$. I define Ω to be the space of all the sets of matchings; that is $\Omega = \{\boldsymbol{\omega} : \boldsymbol{\omega} = (\omega_{1\sigma(1)}, \ldots, \omega_{M\sigma(M)})\}$. In the previous expression $\sigma : \{1, \ldots, M\} \rightarrow \{0, 1, \ldots, m\}$ is a surjective function, with the further requirement that there exists a subset S of cardinality m in $\{1, \ldots, M\}$ such that $\sigma : S \rightarrow \{1, \ldots, m\}$ is injective and $\sigma(s) = 0,\ \forall s \in \{1, \ldots, M\} \setminus S$. With this notation, the constraint on the uniqueness of matching is made explicit. Moreover, the cardinality of each $\boldsymbol{\omega} \in \Omega$ is the same.

The aim, now, is to estimate the probability of each generic matching $P(\omega_{ij})$. This is a hard problem, because the matchings can be mutually dependent and the actual dependencies are not known explicitly. However, a simpler task can be solved; namely the estimation of each match given all the others. Indeed, computing $P(\omega_{ij}|\boldsymbol{\omega})$ is easier than calculating $P(\omega_{ij})$, because the dependencies between the matches of a given $\boldsymbol{\omega}$ can be made explicit or computed more easily. However, because the 2D structure is not perfectly extracted, together with matching errors make this information not accessible all the times. Moreover, when considering $P(\omega_{ij}|\boldsymbol{\omega})$, only the matches that

lead to an admissible reconstruction are taken into account. Indeed, if a match does not generate an admissible reconstruction, for example resulting in degenerate reconstructions or outside the volume of the machine, there is no need to consider it, as it is not correct.

I indicate with $F_I(\omega_{ij}|\Theta)$ and $F_O(\omega_{ij}|\Theta)$ the product of the marginals of the variables considered in case the match $\omega_{ij} = (l_i, r_j)$ is, respectively, a correct match or a wrong one:

$$F_I(\omega_{ij}|\Theta) = \prod_{k=1}^{K} f_k^I(\omega_{ij}|\Theta) \tag{4.1}$$

and

$$F_O(\omega_{ij}|\Theta) = \prod_{k=1}^{K} f_k^O(\omega_{ij}|\Theta). \tag{4.2}$$

f_k^I and f_k^O are the marginal pdf's of the $k-$th selected variable in case of a correct or wrong match respectively. The choice of defining a likelihood in this way, that is, considering the marginals of the distributions, even if some functional relation among variables were considered, is due to the observation, in the learning step, that the relations linking variables do not convey additional stronger evidence for the belonging to either of the groups. Computations of probabilities exploiting the functional relations among variables as discussed in the previous chapter were also tried, but they did not provide overall better results in the final methods. The K variables considered are those resulting different in the Kolmogorov-Smirnov tests, as in Table 3.4. The expressions of F_I and F_O differ from those used originally in the work by Botterill *et al.* [20]. Indeed, my likelihoods come from a more detailed modelling and learning step of the variables of the branches than that in [20]. With the parameter Θ, I indicate the set of constraints that are taken into account: connectedness of the structure (if it is available) and uniqueness of matching for both the branches in ω_{ij}. If information about the connectedness of the structure are not available, only the constraint on the uniqueness of match is considered. Finally, the probability of a match $\omega_{ij} = (l_i, r_j)$,

given all the set of constraints is defined using Bayes' theorem:

$$P(\omega_{ij}|\Theta, \boldsymbol{\omega}) = \frac{\pi(\omega_{ij})F_I(\omega_{ij}|\Theta) \cdot \iota(\boldsymbol{\omega})}{\pi(\omega_{ij})F_I(\omega_{ij}|\Theta) + (1 - \pi(\omega_{ij}))F_O(\omega_{ij}|\Theta)}. \tag{4.3}$$

In (4.3), the term $\pi(\omega_{ij})$ indicates a prior probability that a match is correct and it accounts for the physical characteristics of the pruning machine. It takes into account the Epipolar Geometry of the two views, or, equivalently, the depth of the scene. $\pi(\omega_{ij})$ also controls the admissibility of a reconstruction. The average depth of the vines is at 50 cm from the cameras, so $\pi(\omega_{ij}) \sim \mathcal{N}(0.5, 0.01)$, that is, the depth of a reconstruction is assumed to be normal distributed with mean of 50 cm and standard deviation of 10 cm. However, if the depth of a reconstruction falls outside the interval $[40, 60]$cm, it is considered not admissible. $\iota(\boldsymbol{\omega})$ is an indicator variable which equals 1 if ω_{ij} is compatible with $\boldsymbol{\omega}$ and 0 otherwise. The meaning of (4.3) is the following: it is the probability of a selected match being correct within the reconstructed structure defined by a given possible set of matchings $\boldsymbol{\omega}$. In other terms, (4.3) is the posterior probability of a match, given a model. The probability of a set of matching $\boldsymbol{\omega}$ is defined as

$$P(\boldsymbol{\omega}) := \prod_{\omega \in \boldsymbol{\omega}} P(\omega|\Theta, \boldsymbol{\omega}). \tag{4.4}$$

In this setting, Ω becomes a discrete probability space.

4.2.1 A general method to select the correspondences

In this section I describe the common approach to select the correspondences for all the methods proposed in the present chapter.

Let $L(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ be a loss function:

$$L : \Omega \times \Omega \longrightarrow \mathbb{R}^+. \tag{4.5}$$

An example of a loss could be a distance-like function comparing elementwise the two sets of matchings $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$. Once a loss function has been defined,

a method to find the correspondences is to minimize the expected loss:

$$\boldsymbol{\omega}^* = \arg\min_{\boldsymbol{\omega}_S \in \Omega} \mathbb{E}\left(L(\boldsymbol{\omega}_O, \boldsymbol{\omega}_S)\right), \tag{4.6}$$

where $\boldsymbol{\omega}_O$ is the correct set of matchings and $\boldsymbol{\omega}_S$ is the selected one. A large number of method can be interpreted under this framework. As shown in Vapnik [150], the problem in (4.6),with only two groups involved, is a special case of a pattern recognition problem. The solution of (4.6) depends on the chosen loss, but it generally involves the marginal probabilities of belonging to either of the groups. The loss used for my methods is:

$$L(\boldsymbol{\omega}_O, \boldsymbol{\omega}_S) = |\boldsymbol{\omega}_S \setminus \boldsymbol{\omega}_O| + \alpha|\boldsymbol{\omega}_O \setminus \boldsymbol{\omega}_S|. \tag{4.7}$$

The loss in (4.7) counts the wrong selected matches (the term $|\boldsymbol{\omega}_S \setminus \boldsymbol{\omega}_O|$) and penalizes the correct ones that were not selected. $0 < \alpha \leq 1$ is a parameter used to decide how much the not selected correct correspondences are penalized. Because of the equality of $\boldsymbol{\omega}$'s cardinalities, the loss (4.7) can be rewritten in the following way:

$$L(\boldsymbol{\omega}_O, \boldsymbol{\omega}_S) = \frac{1+\alpha}{2}|\boldsymbol{\omega}_O \triangle \boldsymbol{\omega}_S|, \tag{4.8}$$

where $|\boldsymbol{\omega}_O \triangle \boldsymbol{\omega}_S|$ is the cardinality of the symmetric difference of the two sets $\boldsymbol{\omega}_O$ and $\boldsymbol{\omega}_S$. As the symmetric difference is a set distance, the loss used is proportional to a distance. It can be shown in general that, in this framework, the solution is the following:

$$\text{Assign a match } \omega \text{ to } \boldsymbol{\omega}^* \text{ if: } P(\omega) > \tau, \tag{4.9}$$

where τ is a threshold value that depends on the distance (or loss function) used in the problem. With the specific loss in (4.8), the criterion becomes:

$$\text{Assign a match } \omega \text{ to } \boldsymbol{\omega}^* \text{ if: } P(\omega) > \frac{1}{1+\alpha}. \tag{4.10}$$

However, as said above, the optimality criterion (4.10) involves the marginal probability of the match ω to be correct. As a consequence, a way to compute

the marginals, given the probabilities in (4.3), is needed.

4.2.2 Gibbs Sampling

To compute the marginal probabilities of the matches, a Gibbs sampling algorithm is used. For a detailed description and analysis of the method, I refer to Robert and Casella's book [125]. The Gibbs sampling algorithm is ideal in settings like mine where the *full conditionals*, that is the quantities $P(\omega_{ij}|\Theta, \boldsymbol{\omega})$, are known.

A set of correspondences $\boldsymbol{\omega}$ is encoded as a matrix $W(\boldsymbol{\omega}) \in \mathbb{R}^{M\times m}$ of Bernoulli random variables. Each component $w_{ij}(\boldsymbol{\omega})$ is equal to 1 if the match (l_i, r_j) is in the set of correspondences $\boldsymbol{\omega}$. The probability of success of those Bernoulli random variables are given by (4.3). The constraints

$$\forall j = 1, \dots, |R|, \ \sum_{i=1}^{|L|} \omega_{ij} \leq 1 \tag{4.11}$$

and

$$\forall i = 1, \dots, |L|, \ \sum_{j=1}^{|R|} \omega_{ij} \leq 1 \tag{4.12}$$

are imposed, so that at most one matching for each branch is possible. With the described representation, the $i-$th row $\mathbf{w}_i(\boldsymbol{\omega})$ of the matrix $W(\boldsymbol{\omega})$ records the possible matching of the branch l_i. Analogously, the $j-$th column $\mathbf{w}^j(\boldsymbol{\omega})$ records the matchings for the branch r_j. Thanks to this representation, the mutual compatibility of matchings can be easily verified and not admissible sets of correspondences can be discarded immediately.

The initialization of the algorithm is given by a random admissible set of correspondences. Then, at each iteration t of the Gibbs sampling, given a sequence from the step $t-1$, $\forall i = 1, \dots, |L|$ and $\forall j = 1, \dots, |R|$, the random variables

$$(l_i, r_j)^t \sim Be\left(P((l_i, r_j)|\{(l_k, r_h)^t\}_{k,h}, \{(l_K, r_H)^{t-1}\}_{K,H})\right), \quad \text{with } k < i, h < j, K > i, H > j \tag{4.13}$$

are generated. In the previous expression, $\{(l_k, r_h)^t\}_{k,h}$ is the set of all match-

ings, updated to step t, with indices less than i and j, whereas $\{(l_K, r_H)^{t-1}\}_{K,H}$ is the set of all matchings, updated to the $(t-1)-$th step, with indices greater than i and j.

Now, the marginal probabilities can be easily estimated thanks to the law of large numbers:

$$P(\omega_{ij}) = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} w_{ij}(\boldsymbol{\omega}^{\boldsymbol{\tau}}). \tag{4.14}$$

In the actual computation of the marginal probabilities, 5000 iterations are considered, thus leading to the approximation:

$$P(\omega_{ij}) \approx \frac{1}{5000} \sum_{\tau=1}^{5000} w_{ij}(\boldsymbol{\omega}^{\boldsymbol{\tau}}). \tag{4.15}$$

4.2.3 Bayes classifier method

The first proposed method that I explain is the Bayes classifier method, first proposed in Botterill *et al.* [20]. The name comes from the fact that the basic probabilities used to compute the correctness of a match are given simply through Bayes' theorem, that is formula (4.3). This is the simplest of the methods proposed, as it is the straightforward application of the framework and Gibbs sampling previously explained.

When applying the algorithm, all of the three cameras of the pruning machine are used in the following way. If the main orientation of a branch is horizontal, then the correspondences are sought using the right and top cameras of the system. Instead, if the orientation is mainly vertical, correspondences are sought by using the right and left cameras. This usually helps in decreasing admissible matches and therefore ambiguities.

4.3 Support Vector Machine

In the learning process seen in Chapter 3, it was found that often the distribution of the variables considered overlap and it is therefore difficult deciding to which group a given match belongs. However, it may happen that mapping the input data into a higher-dimensional space (called *feature space*) could give a more definite criterion to judge the correctness of a correspondence. For this reason, a Support Vector Machine is constructed. For

a detailed explanation of the Support Vector Machine, I refer to Vapnik's book [150] and to Hastie's *et al.* [68]. The choice of a Support Vector Machine algorithm is due, mainly, to two important features that this class of algorithms show. On one side the ability of fitting a regularised model of arbitrary complex multi-dimensional data; on the other side the tolerance to some misclassified training data.

A Support Vector Machine model for classification is built considering only the geometrical variables. This choice follows from the fact that the geometrical variables resulted being more discriminative between the two groups. Some other variables of the branches are then integrated in a subsequent step.

Linear, polynomials and Gaussian kernels are considered when building the optimal Support Vector Machine model. The ground truth data are randomly divided into two different sets, a training set with 800 data points and a test set with 106 data points. Half of the data in the training and testing set are from the group of correct correspondences, while, the other half from the group of wrong ones. For each kernel, a Support Vector Machine model is built using the training set and tuned over a grid of supplied ranges of the parameters. The parameters to be tuned are: the cost of misclassification (for all of the three kernels), the coefficient γ (i.e. the reciprocal of the variance in polynomial and Gaussian kernels) and finally the degree d and the constant term of the polynomial kernel. The described procedure has been repeated for 5000 times. The classifiers constructed are then compared by performances on the test set using the misclassification error. Finally, for each of the kernels, the best classifier is retained. The kernel showing the best average performance overall is the linear one, with about 27% of misclassification error on the test set. The polynomial and Gaussian kernels have very close performances to each other, with average misclassification error of about 42%. The best linear Support Vector Machine classifier has a misclassification error of 23% and of 12% specific on the set of correct correspondences.

From the theory, see for example Vapnik [150], it is known that the group of a new datum can be predicted using a decision function that exploits only the products of the support vectors with the new datum. Therefore, indi-

cated with $\mathcal{S}$ the set of support vectors and with y_i a dummy variable that is 1 if the support vector i belongs to the group of correct correspondences and -1 otherwise, the obtained decision function is of the type:

$$F(\mathbf{x}) = \text{sign}\left(\sum_{\mathbf{x}_i \in \mathcal{S}} y_i \alpha_i \mathbf{x} \cdot \mathbf{x}_i + b\right), \tag{4.16}$$

where α_i and b are parameters computed in the construction of the Support Vector Machine classifier (α_i's are the Lagrange multipliers computed when constructing the classifier and b the constant term of the optimal hyperplane).

The information about the predicted group of a new datum have to be integrated with the probability of a match being correct considering the other variables of a branch. In order to do this, the decision function has to be used to compute the probability of a new given match to be correct. Predictions from support vector machine are not probabilistic. To overcome this disadvantage, there have been proposed approaches similar to support vector machine that can provide probabilistic outputs. Examples are provided by Tipping [143] or Menor *et al.* [107]. However, in my case those approaches are not suitable, because information on the distribution of generic matchings $P(\omega_{ij})$ should be available. However, as I explained in section 4.2, it is not the case. Instead, in order to obtain probabilistic outputs, I follow the approach described in Platt [122]. Let $f(\mathbf{x})$ be the term inside the sign function in (4.16). The probability of a new match being correct is modelled with a sigmoid function:

$$P((l,r) \text{ is correct}|\Theta, \boldsymbol{\omega}, f) = \frac{1}{1 + \exp\left(Af(\mathbf{x}) + B\right)}, \tag{4.17}$$

where A and B are constant to be determined and $f(\mathbf{x})$ is the function f evaluated in the vector $\mathbf{x}$ of the geometrical variables of the new match (l,r). I refer to Platt [122] for a discussion of the hypothesis leading to parametrization (4.17). The parameters A and B of (4.17) are fit using maximum likelihood estimation given a training set. The training set (f_i, t_i) is defined in the following way: $f_i = f(\mathbf{x}_i)$ and t_i's are defined as $t_i = 1$ if the

$i-$th datum is a correct correspondence and $t_i = 0$ otherwise. Finally, set

$$p_i = \frac{1}{1 + \exp\left(Af_i + B\right)}, \tag{4.18}$$

where A and B are found minimizing the negative log-likelihood of the training data

$$\min\left\{-\left(\sum_i t_i \log(p_i) + (1 - t_i)\log(1 - p_i)\right)\right\}. \tag{4.19}$$

Platt [122] states that this procedure is suitable only for Support Vector

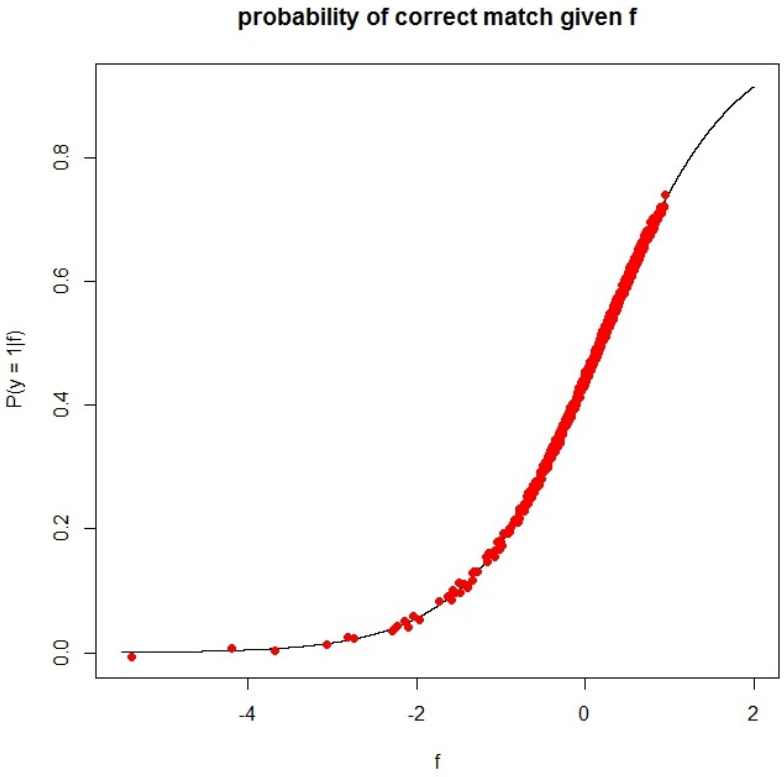


Figure 4.1: Fitted probability function for the model in (4.17). The continuous line is the function assigning to a given vector $\mathbf{x}$ of geometrical variables (in reality, to $f(\mathbf{x})$) coming from an admissible correspondence the probability that the match is correct. The red dots represent the actual probability, computed on the training data, that a match is correct.

Machine with linear kernel. The reason is that training a support vector

machine introduces a bias. Indeed, for data that are close to the margins, the corresponding f will be forced to have absolute value equal to 1 (or corresponding values due to the intercept term of $f(\mathbf{x})$), which is not a common value for test examples. However, for linear support vector machines, the bias introduced is small, because for an input of dimensionality p, a maximum of $p+1$ support vectors generally lie on the margin and this is usually a small fraction of the training data. Also, the optimal performance for linear support vector machine is reached for small misclassification costs, which causes the bias on the margin failures to be small. In other cases, the bias introduced is more complex to be handled. Usually, approaches based on cross validation are used.

The estimated values are $A = -1.498522$ and $B = -0.866753$. Figure 4.1 shows the sigmoid function fit to the training data probabilities. It can be seen that the estimated function fits well to the training data. For the optimization, the whole set of data was used with the decision function given by the best linear Support Vector Machine model.

Once the model (4.17) has been estimated, it has to be integrated with the information coming from the other variables that were not considered in the construction of the classifier. Therefore, following (4.3), the probability under the Support Vector Machine model $P_{SVM}(\omega_{ij}|\Theta, \boldsymbol{\omega})$ of a match being correct, given a set of correspondences $\boldsymbol{\omega}$, is defined as

$$\frac{\pi(\omega_{ij})F_I(\omega_{ij}|\Theta)P(\omega_{ij}|\Theta, \boldsymbol{\omega}, f) \cdot \iota(\boldsymbol{\omega})}{\pi(\omega_{ij})F_I(\omega_{ij}|\Theta)P(\omega_{ij}|\Theta, \boldsymbol{\omega}, f) + (1-\pi(\omega_{ij}))F_O(\omega_{ij}|\Theta)(1-P(\omega_{ij}|\Theta, \boldsymbol{\omega}, f))}. \tag{4.20}$$

In (4.20) I have indicated with $F_I(\omega_{ij}|\Theta)$ the product of the likelihoods of the remaining variables taken into account in the model in case the match is correct and analogously for $F_O(\omega_{ij}|\Theta)$. In this case, the other variables considered are: maxima and minima of the branches, connectedness of the structure, endpoints agreement, back-projection error and overlapping. I remind that Table 3.4 provides a summary of the variables used in the different methods.

The method here presented, then, adopts the same Gibbs sampling algorithm explained in the previous section to estimate the marginal probabilities, exactly as the Bayes classifier.

4.4 Gradient Boosting Method

In this section, I explain the construction of a model that uses all the variables and boosts simple learners through a functional gradient descent algorithm. Since their birth, boosting procedures have been successfully applied to many problems in statistics and machine learning. Two remarkable results are given by a method based on gradient boosting to help detecting the Higgs' boson [81] or the victory in the 2015 Microsoft Malware Classification Challenge [82]. Boosting algorithms start with sensible estimators or classifiers, called *learners*, and iteratively try to improve their performances on a training set of data. For the model I built, I used a functional gradient descent algorithm to improve the performances of the base learner. Detailed analysis and description of boosting techniques and application can be found, for example, in Buhlmann and Yu [24], Hastie *et al.* [68], Friedman [55] and Friedman and Popescu [56]. In the next explanation, I follow Buhlmann and Yu [24]. This algorithm has been preferred over other common methods, for example random forests, because of its proved outstanding prediction power, as shown, for example, in the references above. A tree-based classification approach was also experimented on my problem; however it showed a higher misclassification error and it provides decision functions that are more difficult to treat.

The main task is to estimate a function $F : \mathbb{R}^d \to \mathbb{R}$ that minimizes an expected cost

$$\mathbb{E}\left(C(Y, F(X))\right), \; C(\cdot,\cdot) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+ \tag{4.21}$$

given data (Y_i, X_i), with $i = 1, \ldots, n$. X denotes a $d-$dimensional predictor variable and Y is the response variable that can be continuous for regression problem or discrete for classification. In my case, $Y \in \{0, 1\}$, with $Y = 1$ if a match belongs to the set of correct correspondences. The estimation of F can be carried out through a constrained minimization of the empirical risk

$$\frac{1}{n}\sum_{i=1}^{n} C(Y_i, F(X_i)) \tag{4.22}$$

by applying a functional gradient descent algorithm. To assure a good convergence rate of the algorithm, the cost C is supposed to be smooth and

convex in the second argument. The minimizer of (4.22) is imposed to satisfy a constraint in terms of an additive expansion of *simple* learners $h(x, \hat{\theta}),\ x \in \mathbb{R}^d$, where $\hat{\theta}$ is an estimated parameter. For my specific problem, I used $C(y, f) = \frac{1}{2}(y-f)^2$ as cost function, whereas the simple learners are of the type $h(x, \hat{\theta}) = \frac{1}{2}\log(\frac{p(x|\hat{\theta})}{1-p(x|\hat{\theta})})$, where $p(x|\hat{\theta})$ is the probability of a match being correct. The algorithm goes as follows.

Step 1. Given the data (Y_i, X_i), the first fitted learner is

$$\hat{F}_0(x) = h(x, \hat{\theta}_{Y,X}).$$

$\hat{\theta}_{Y,X} = \arg\min_\theta \sum_{i=1}^n (Y_i - h(X_i, \theta))^2$ and $m = 0$.
Step 2. The negative gradient vector of the cost function is computed and evaluated at the current $\hat{F}_m(\cdot)$:

$$U_i = -\frac{\partial C(Y_i, F)}{\partial F}\Big|_{F=\hat{F}_m(X_i)},\ i = 1, \ldots, n.$$

Then, the simple learner is fitted to the gradient vector

$$\hat{f}_{m+1}(x) = h(x, \hat{\theta}_{U,X})$$

with $\hat{\theta}_{U,X} = \arg\min_\theta \sum_{i=1}^n (U_i - h(X_i, \theta))^2$.
Step 3. Search for the best step size

$$\hat{w}_{m+1} = \arg\min_w \sum_{i=1}^n C(Y_i, \hat{F}_m(X_i) + w\hat{f}_{m+1}(X_i))$$

and update:

$$\hat{F}_{m+1}(\cdot) = \hat{F}_m(\cdot) + \hat{w}_{m+1}\hat{f}_{m+1}(\cdot).$$

Step 4. m is increased by one and steps 2. and 3. repeated.

The procedure here described is called L_2Boosting, because of the particular cost function used. Thus, L_2Boosting is a repeated least square fitting of residuals, cfr. Buhlmann and Yu [24] or Friedman [55]. For the classification problem with two classes, analogously to the Support Vector Machine, the

decision function provided by the algorithm is $\text{sign}(\hat{F}_m(x))$.

The L_2Boosting is applied to all the variables together. However, when building the model $\hat{F}_m(\cdot)$, only one predictor variable is chosen at each iteration. Therefore, at each step of the algorithm, the couple

$$(\hat{\theta}_{H,X}, \hat{\iota}) = \arg\min_{\theta,\iota} \sum_{i=1}^{n} (H_i - h(X_i^{\iota}, \theta))^2$$

is found, where $H = Y, U$ and X_i^{ι} is the component ι of the $i-$th datapoint. This allows the functional gradient descent algorithm to fit a model in an additive way and to do variable selection at the same time, so that the most relevant variables are automatically selected.

In Figure 4.2, the trajectories of the weights and variables selected are shown. In total, 200 iterations of the functional gradient descent algorithm were executed. The best model was found at the 58th iteration using Akaike's Information Criterion (AIC). Moreover, the algorithm gives an indication on the importance of the variables. The thickness is the first chosen variable discriminating among the groups. Other important variables are the second and third eigenvalues of the derivatives of the branches. In Figure 4.2 it seems that only three variables were selected, but maxKinkAngle, curvature and the angle between $\vec{\mathbf{d}}$ and $\vec{\mathbf{j}}$ were selected with small weights.

Finally, there is one more step to check for the soundness of the model built with the above procedure: testing for interaction terms between the model. In fact, the model constructed is an additive one, but there may be more terms that take into account higher order interactions between the variables. I used the Friedman's test as described in Friedman and Popescu [56]. Let the set of predictor variables be split into two disjoint subset C and S, so that the vector of predictor variables can be written as $X = (X_C, X_S)$. The *partial dependence of the function* $F(x)$ on X_S is defined as

$$F_S(X_S) = \mathbb{E}_{X_C}(F(X_C, X_S)). \tag{4.23}$$

Partial dependence functions can be estimated from the data by

$$\hat{F}_S(x_S) = \frac{1}{n} \sum_{i=1}^{n} \hat{F}(x_S, x_C^i). \tag{4.24}$$

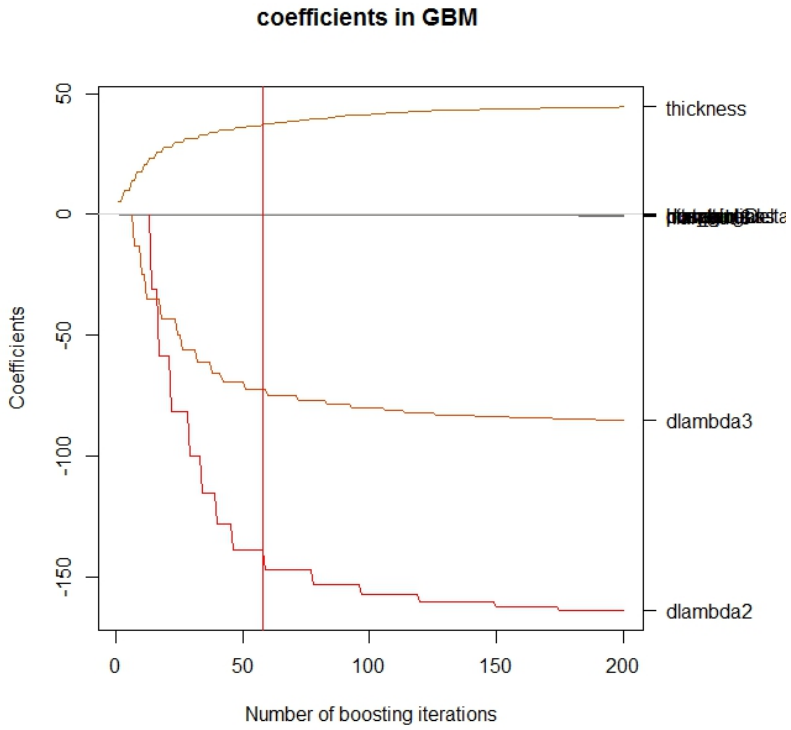


Figure 4.2: Trajectories of the variables and weights selected by the functional gradient descent algorithm. The jumps in the trajectories mean that the weight corresponding to that variable has been updated. The red vertical line indicates the iteration corresponding to best model, selected via AIC. The other variables either are not chosen or have very small weights; this cause their name to overlap around the x axis. Here, "dlambda2" and "dlambda3" are the second largest and the third largest eigenvalues of the "derivatives" of branches.

If two variables x_j and x_k do not interact, then the partial dependence of $F(x)$ on $x_S = (x_j, x_k)$ can be decomposed as

$$F_{jk}(x_j, x_k) = F_j(x_j) + F_k(x_k). \tag{4.25}$$

Therefore, because of (4.24) and (4.25), to test for the presence of interactions between two variables (x_j, x_k), the statistic:

$$H_{jk}^2 = \sum_{i=1}^{n} \left[\hat{F}_{jk}(x_j^i, x_k^i) - \hat{F}_j(x_j^i) - \hat{F}_k(x_k^i)\right]^2 \Big/ \sum_{i=1}^{n} \left[\hat{F}_{jk}(x_j^i, x_k^i)\right]^2 \tag{4.26}$$

can be used. It measures the fraction of variance of $\hat{F}_{jk}(x_j, x_k)$ not captured by $\hat{F}_j(x_j) + \hat{F}_k(x_k)$ over the data distribution. If variables j and k have no interactions, H_{jk}^2 will have a value close to zero. The test here described, applied to the model built with the functional gradient descent revealed no interactions among the selected variables.

Once the predictive model $\hat{F}$ has been estimated, it has to be used to compute the probability that a match is correct and finally integrated with the probabilities coming from other variables, such as the overlapping or the back-projection error. The probability of a match being correct, given the predictive model $\hat{F}$ is:

$$P((l, r) \text{ is correct}|\Theta, \boldsymbol{\omega}, \hat{F}) = \frac{1}{1 + \exp\left(-2\hat{F}(x)\right)}. \tag{4.27}$$

Analogously to what has been done with the Support Vector Machine method, the probability under the Gradient Boosting model $P_{GBM}(\omega_{ij}|\Theta, \boldsymbol{\omega})$ of a match being correct, given a set of correspondences $\boldsymbol{\omega}$, is computed as:

$$\frac{\pi(\omega_{ij}) F_I(\omega_{ij}|\Theta) P(\omega_{ij}|\Theta, \boldsymbol{\omega}, \hat{F}) \cdot \iota(\boldsymbol{\omega})}{\pi(\omega_{ij}) F_I(\omega_{ij}|\Theta) P(\omega_{ij}|\Theta, \boldsymbol{\omega}, \hat{F}) + (1 - \pi(\omega_{ij})) F_O(\omega_{ij}|\Theta)(1 - P(\omega_{ij}|\Theta, \boldsymbol{\omega}, \hat{F}))}, \tag{4.28}$$

with the same meaning for the symbols as in (4.20). The marginals probabilities are computed using the Gibbs sampling algorithm.

4.5 Averaging the methods that use Gibbs sampling

Having the Bayes classifier, Support Vector Machine and Gradient Boosting models, a new method can be proposed by simply averaging them in a suitable way. All of the methods using Gibbs sampling attempt to compute, in general, $P((l, r)|\boldsymbol{\omega})$, where, again, $\boldsymbol{\omega}$ is a set of matchings. The quantity

$P((l,r)|\boldsymbol{\omega})$ can be re-written as:

$$P((l,r)|\boldsymbol{\omega}) = \sum_{i=1}^{3} P((l,r)|\mathcal{M}_i, \boldsymbol{\omega}) P(\mathcal{M}_i|\boldsymbol{\omega}), \tag{4.29}$$

where $\mathcal{M}_i$ is the $i-$th model. For example, in my case, $\mathcal{M}_1$ = Bayes classifier, $\mathcal{M}_2$ = support vector machine and $\mathcal{M}_3$ = gradient boosting. The terms $P((l,r)|\mathcal{M}_i, \boldsymbol{\omega})$'s are all known, but the converse is true for $P(\mathcal{M}_i|\boldsymbol{\omega})$'s, which is the probability the model $\mathcal{M}_i$ is correct if the set of matching $\boldsymbol{\omega}$ is observed. However, in the present context, the terms $P(\mathcal{M}_i|\boldsymbol{\omega})$'s cannot be known. If they were, then the marginal probability $P(\boldsymbol{\omega})$ of a set of matching $\boldsymbol{\omega}$ would be known directly. However, this is not feasible, because, as it was said in a previous section, it is a very difficult task.

The problem of understanding $P(\mathcal{M}_i|\boldsymbol{\omega})$ can be overcome by thinking that the terms $P(\mathcal{M}_i|\boldsymbol{\omega})$'s simply represent a probability on a finite space of models. Moreover, if a set of matching is randomly chosen, very little *a priori* informations is available to decide if a model is more correct than another one. Therefore, $P(\mathcal{M}_i|\boldsymbol{\omega})$ can be taken to be the same for all i's. In this case, a new method is obtained, that seeks for majority of consensus among the classifiers. It is known, e.g. Breiman [22], Hastie *et al.* [68] or the introduction in Freund *et al.* [54], that averaging methods can result in more stable procedures if the original methods provide discordant decisions. The algorithm proposed here could work well if the Bayes classifier, Support Vector Machine and Gradient Boosting are complementary to each other. That is, if the three classifiers find different sets of correspondences and give them non-zero probabilities that summed together account for a value greater than 1/2, such a method would be effective. On the other hand, if two or more misclassify a correct correspondence, there is no procedure that can amend that error, also because understanding the a priori true correctness of a matching is very difficult on matches that do not come from ground truth.

With the method here presented, the probability of a match being correct, given a set of matching is defined as

$$P_{AV}(\omega_{ij}|\Theta, \boldsymbol{\omega}) = \frac{1}{3}\left(P_{BC}(\omega_{ij}|\Theta, \boldsymbol{\omega}) + P_{SVM}(\omega_{ij}|\Theta, \boldsymbol{\omega}) + P_{GBM}(\omega_{ij}|\Theta, \boldsymbol{\omega})\right) \tag{4.30}$$

and then used in the Gibbs sampling algorithm to compute the marginal probabilities.

Chapter 4 presents a first class of methods proposed to solve my problem, namely methods that compute the probability of a match being correct. It is a first step towards achieving the fourth objective in the list of Section 1.4. Moreover, the algorithms presented here are original contributions to solving correspondence problems in the case of branched structures.

Chapter V

Proposed methods for computing correspondences

5.1 Introduction

In this chapter I describe another class of methods used to find correspondences: a maximum likelihood algorithm, RANSAC and a graph matching algorithm. The methods here presented have been grouped together because they directly compute or select correspondences.

My main contribution in this chapter is the implementation and adaptation of state-of-the-art algorithms to my specific problem, together with a comparison between the different methods.

This chapter is organized as follows. In the next section I describe a maximum likelihood algorithm method. After that, I proceed to a description of RANSAC and how it has been applied to my problem. Finally, I conclude with the explanation of a graph matching approach to vine correspondences.

5.2 Maximum likelihood method

In this section I explain an adaptation of a maximum likelihood algorithm to my problem. Recalling the notation introduced in the previous chapter, let L and R be the left and right images. Let $|L|$ and $|R|$ the number of branches seen in L and R, respectively. $\omega_{ij} = (l_i, r_j)$ indicates the matching between two generic branches l_i and r_j, with $i = 1, \ldots, |L|$ and $j = 1, \ldots, |R|$.

As noted in the previous chapter in Section 4.2, computing the marginal probability $P(\omega_{ij})$ of a generic matching ω_{ij} is a complex problem. The same is true for the likelihoods of matchings. However, again, by conditioning on other matches and exploiting suitable constraints provided by the particular structure of the problem, an easier task can be addressed.

In the same notation of the previous chapter, I indicate with $F_I(\omega_{ij}|\Theta)$ and $F_O(\omega_{ij}|\Theta)$ the product of the marginals of the variables considered in case the match $\omega_{ij} = (l_i, r_j)$ is, respectively, a correct match or a wrong one:

$$F_I(\omega_{ij}|\Theta) = \prod_{k=1}^{K} f_k^I(\omega_{ij}|\Theta) \tag{5.1}$$

and

$$F_O(\omega_{ij}|\Theta) = \prod_{k=1}^{K} f_k^O(\omega_{ij}|\Theta). \tag{5.2}$$

f_k^I and f_k^O are the marginal pdf's of the $k-$th variable in case ω_{ij} is a correct or wrong match respectively. See Table 3.4 for a list of the variables used in the maximum likelihood algorithm. As in (4.3) and because Ω is a discrete space, the likelihood of a match being correct is defined as

$$P(\omega_{ij}|\Theta, \boldsymbol{\omega}) = \frac{\pi(\omega_{ij})F_I(\omega_{ij}|\Theta) \cdot \iota(\boldsymbol{\omega})}{\pi(\omega_{ij})F_I(\omega_{ij}|\Theta) + (1 - \pi(\omega_{ij}))F_O(\omega_{ij}|\Theta)}.$$

The symbols have the same meaning as in (4.3).

To select the maximum likelihood solution, the Gibbs sampling algorithm is used. However, there is no need to compute the marginal probabilities; instead, the matches being sampled more often are the ones considered correct and therefore retained.

5.3 RANSAC

In this section I describe how the RANSAC algorithm is applied to find correspondences between branches. Many modifications of RANSAC have been proposed in the past, but I adopt the classic formulation as first defined in Fischler and Bolles [51].

Each iteration of the algorithm is divided into two steps. The first one is sometimes referred to as *hypothesis generation* step. Among the matches that lead to an admissible reconstruction, m matches are randomly chosen. I want to remark that admissible matches satisfy the depth constraints of the system and generate admissible reconstructions. From the m initially selected matches a model for the vines is learnt in the following way. For the

RANSAC algorithm, only the geometrical variable resulting different from the Kolmogorov-Smirnov test are considered, see Table 3.4. The values of those selected variables of the reconstructed branch are recorded and their distributions learnt independently. Given those distributions, the extremes of an empirical 95% confidence interval $[L, U]$ are computed for each variable:

$$L = \bar{x} - q^l SE,\ U = \bar{x} + q^u SE, \tag{5.3}$$

where $\bar{x}$ is the sample mean of the distribution, q^l and q^u are the 0.025 and 0.975 quantiles respectively and SE is the standard error: $SE = \sqrt{\hat{\sigma}^2/m}$. The number m of selected matches depends on the number of branches detected in the images. m has to be large enough to compute parameters for the distributions of the variables and, because of the discussion made at the beginning of Section 4.2, $m < \min\{|L|, |R|, |T|\}$. In consideration of this, the computation of the parameters of the distributions, and the fact that there is no *a priori* informations about the number of canes in an image, I defined m to be $\lfloor \frac{\min\{|L|,|R|,|T|\}}{2} \rfloor$, which is still a small value if compared to the total number of matching candidates. Experimentally, the chosen value is acceptable and, on the observed cases, about 10 or 11. Lower values of m could not assure a significant estimation of L and U. This results in a very high rejection rate or biased estimations. Conversely, higher values for m cause RANSAC to fail, because the probability of finding an outlier-free hypothesis set is very small. Other values for m have been tried, but the choice made seems a good trade-off between a good estimation of the model and sampling outlier-free hypothesis sets.

The second step of RANSAC algorithm is sometimes referred to as *verification step*. In this part of the algorithm, RANSAC checks which new elements are consistent with the model instantiated in the hypothesis generation step. To decide which other matches are considered inlier for the model, I use the confidence intervals defined above. If a new match has every variable falling in the specific confidence interval, it is considered consistent with the model instantiated. If the distribution of the variables is symmetric, then this is equivalent to having a distance-based criterion, such as the one described in Hartley and Zisserman [67]. If $\mathbf{x}^M$ indicates the vector contain-

ing the mean of the variables considered, using the confidence interval is the same as considering a new match to be correct if

$$\text{dist}^2(\mathbf{x}, \mathbf{x}^M) < r^2,$$

where $\text{dist}^2(\mathbf{x}, \mathbf{x}^M)$ is defined as

$$\prod_i \frac{(x_i - x_i^M)^2}{SE_i^2}$$

and r^2 is then equal to $\prod_i q_i^2$. The index i means that the quantities are relative to the $i-$th variable and q_i indicates the 0.975 quantile. This criterion is actually slightly more restrictive than the one suggested in [67]. Moreover, when deciding if a new match is an inlier, the constraint on the uniqueness of matchings is still considered.

The two previous steps are iterated 5000 times for each image and the model with the largest number of inliers retained. It is the only number of iterations considered, because a larger number only makes RANSAC slower, without much improvement in its performances.

In the hypothesis generation step, a condition on the connectedness of the structure can be added. The pairwise distances between the selected branches can be computed. Then, the probability of observing the minimum value can be computed through (3.7). If the resulting probability is less than 0.05, the structure is considered disconnected. However, computing this probability is an expensive operation and slows RANSAC noticeably. Moreover, after having observed that the provided solution with the connectedness constraint is not of significantly better quality, this further check was removed.

5.4 Tree matching

In this section I explain a tree matching algorithm applied to corresponding vine structures. The following approach differs from all the others methods that I propose, because it looks for correct correspondences avoiding the computation of the three dimensional reconstruction of the branches. The method discussed here follows the work by Kumar *et al.* [91]. As already

seen in Chapter 3, the structures I deal with can naturally be described as a tree in a graph theoretical sense. Therefore, tree matching can provide a suitable framework for my correspondence problem.

The first element that needs to be defined is a suitable cost of matching two nodes. This term has to take into account a cost related to the branches selected and their parents to encourage connectedness. I indicate the vector of attributes of the $i-$th and $j-$th branch in the left and right image respectively as $\mathbf{a}_L^i$ and $\mathbf{a}_R^j$ and with $p(i)$ the ancestor of the $i-$th branch in the image it belongs to. Then, a simple expression for a cost is

$$c_{ij} = c((l_i, r_j)) = \left(\mathbf{a}_L^i - \mathbf{a}_R^j\right)^T \Sigma^{-1} \left(\mathbf{a}_L^i - \mathbf{a}_R^j\right) + \left(\mathbf{a}_L^{p(i)} - \mathbf{a}_R^{p(j)}\right)^T \Sigma^{-1} \left(\mathbf{a}_L^{p(i)} - \mathbf{a}_R^{p(j)}\right). \tag{5.4}$$

In (5.4), Σ is the covariance matrix for the variables in the vectors $\mathbf{a}$'s. The assumption behind the cost in (5.4), as in Cho *et al.* [27], is that there is a relation such as $\mathbf{a}_R^j = \mathbf{a}_L^i + \boldsymbol{\varepsilon}$ if the branch l_i correctly corresponds to r_j. $\boldsymbol{\varepsilon}$ is an error term: $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. The matrix Σ is estimated using the sample covariance $\hat{S}$ of the variables considered in the ground truth correspondences. Even if this assumption was correct, it is likely that the self-similarity of the vines distorts the matchings. The structure extraction step can also have a great influence in the performances of this method. If the information of the parents are not available, a penalization is given to the term involving the ancestors of a branch. This criterion of matching is more flexible than strictly imposing a rigid ancestry requirement, because matches breaking that constraint are penalised, but still matchable.

Now, the tree matching problem is to find the lowest cost mapping that preserves ancestry. Indicated with T_L and T_R the tree structures in the two images, let G be a complete bipartite graph between $T_L \cup \{N_L\}$ and $T_R \cup \{N_R\}$, where N_L and N_R are auxiliary no-match nodes. The goal of the algorithm is to find a set of edges $M \subset G$ such that the total cost $c(M) = \frac{1}{M} \sum_{i,j} c_{ij}$ is minimized, where $(l_i, r_j) \in M$. Finally, to view this approach in a graph matching framework, it is enough to observe that the matrix $\mathbf{W}$ in (2.9) is simply the one having the costs c_{ij} as elements and the vector $\mathbf{x}$ has a 1 if the corresponding edge is in M and 0 otherwise.

To find the optimal set of edges M^*, a random-walk Metropolis Hastings algorithm is used. For a complete discussion of the random-walk version of the Metropolis Hastings algorithm, I refer again to Robert and Casella's book [125]. An objective function is defined as

$$f(M) = \exp\left(-c(M)\right). \tag{5.5}$$

At each iteration of the algorithm, a new set of edges $\hat{M}$ is proposed and becomes the new reference set with probability

$$\alpha\left(\hat{M}|M\right) = \min\left(1, \frac{f(\hat{M})}{f(M)}\right). \tag{5.6}$$

The algorithm runs for $N = 5000$ iterations, as opposed to the 100 iterations indicated in the work by Kumar *et al.* [91], and the set of edges with the lowest cost is returned. The algorithm is initialized with a random set of matchings. In the final set returned, there is a further filter step: only the matchings giving an admissible reconstruction are retained. The number of iterations chosen seems to be sufficient for the convergence of the algorithm. Randomly sampled chains seem to suggest that convergence on the cost of the set of edges is reached after the 2000th iteration and they exhibit a good mixing. To provide further information, the Geweke diagnostic, i.e. a test on the difference of means between the first 0.1 and last 0.5 proportions of the chain, is also implemented in the algorithm. In case of no rejection of H_0 in the Geweke's diagnostic, no correspondences are returned.

The methods presented here constitute another step towards reaching objective 4. in the list of Section 1.4. The discussed algorithms directly compute correspondences between the images and they are modifications of state-of-the-art procedures. In their development, these methods needed a careful customization in order to be applied to my problem, so that, as in the case of the tree matching algorithm, they can be considered generalizations of previously proposed methods.

Chapter VI

Results

6.1 Introduction

In this chapter I present the results coming from the application of the methods proposed to images showing challenging branched structures.

First, I begin with precision - recall results on a set of ground truth data that was not used to build the algorithms used to find the correspondences. This dataset is relatively small, consisting of three plants. However, it shows a great degree of complexity in the structure of the vines and the number of branches is also large, if compared to the dataset used to train the models.

Subsequently, the methods are tested on 6 large datasets and a smaller one, corresponding to entire rows of different varieties of grapevines, coming from different vineyards. In these datasets, the plants used to train the models are included, even though they are only 6 plants and they are not the same images. The analysis of the performances of the methods on these datasets focus on two main aspects: on one hand the number of the correspondences judged correct by the methods; on the other hand the quality of the final 3D structures provided by the methods. I will define these quantities later in the chapter.

6.2 Precision - recall results

Thanks to the ground truth data, precision and recall can be easily computed for each method. Precision and recall of maximum likelihood, RANSAC and tree matching do not depend on the parameter α in (4.10), whereas the others do. In Figure 6.1, the precision - recall results for the different methods are shown. In Table 6.1 there are the corresponding values.

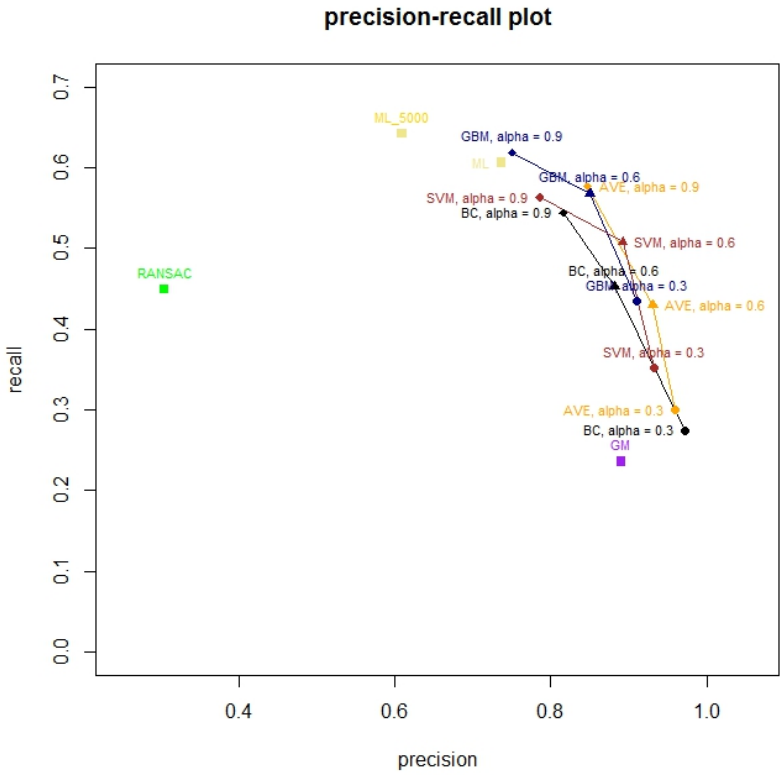


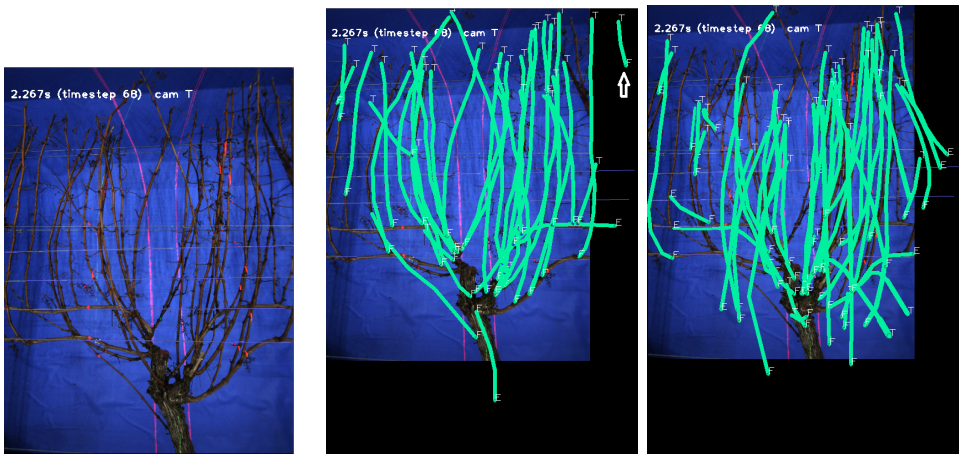
Figure 6.1: Precision - recall plot for the proposed methods. The methods using Gibbs sampling show the best performances overall, with Gradient Boosting (GBM) and Support Vector Machine (SVM) doing better overall. The averaged method (AVE) and Bayes classifier (BC) are rather similar. Among the methods not depending on the parameter α, maximum likelihood showed surprisingly good results on the ground truth data. Finally, RANSAC and the tree matching algorithm showed lower performances than the other methods.

The group of algorithms depending on the parameter α show overall better performances than RANSAC and tree matching. The maximum likelihood algorithm performs surprisingly well on the ground truth data.

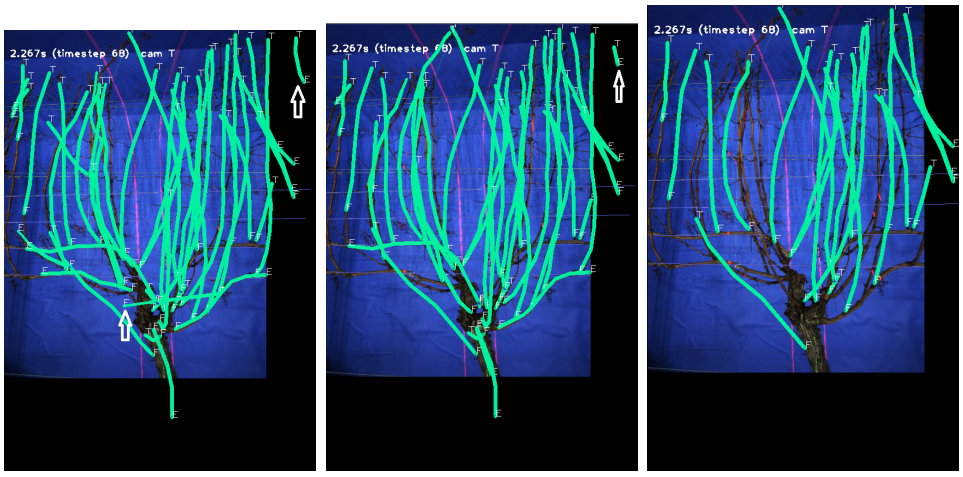
In Figure 6.2a, an example of a vine structure used to compute the precision recall results is shown. I now show how the different methods compare on the same structure given the same timestep. Sometimes, even for a human eye, understanding which reconstructions are correct is difficult. On the particular vine shown in Figure 6.2a, the tree matching algorithm seems to

method	precision	recall
GBM, $\alpha = 0.9$	0.7510509	0.6186926
GBM, $\alpha = 0.6$	0.8496239	0.5680899
GBM, $\alpha = 0.3$	0.9106450	0.4343744
SVM, $\alpha = 0.9$	0.7861606	0.5632851
SVM, $\alpha = 0.6$	0.8920511	0.5074455
SVM, $\alpha = 0.3$	0.9318227	0.3516147
AVE, $\alpha = 0.9$	0.9594363	0.5764083
AVE, $\alpha = 0.6$	0.9309859	0.4292829
AVE, $\alpha = 0.3$	0.8474832	0.2995187
BC, $\alpha = 0.9$	0.8165299	0.5438092
BC, $\alpha = 0.6$	0.8820593	0.4527062
BC, $\alpha = 0.3$	0.9723559	0.2746188
ML	0.7368567	0.6060023
ML_{5000}	0.6095026	0.6419456
RANSAC	0.305527	0.4488137
GM	0.8902616	0.2357743

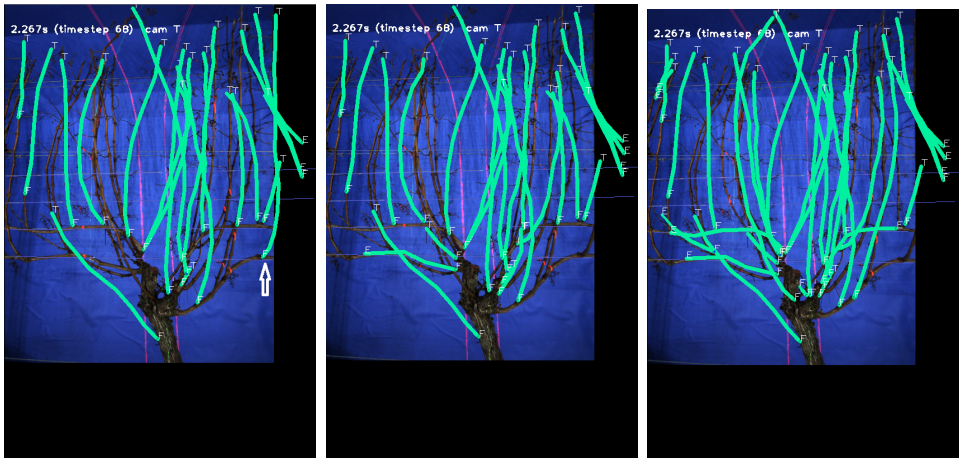
Table 6.1: Precision - recall values for the algorithms. GBM = Gradient Boosting Method, SVM = Support Vector Machine, BC = Bayes Classifier, AVE = Averaged method (GBM, SVM and BC), ML = Maximum Likelihood (200000 iterations), ML_{5000} = Maximum Likelihood (5000 iterations) GM = tree matching.



(a) Vine structure **(b)** tree matching **(c)** RANSAC



(d) maximum likelihood **(e)** ML (5000 it.) **(f)** Bayes classifier



(g) averaged method **(h)** SVM **(i)** gradient boosting

Figure 6.2: Comparison of the correspondences found and reconstructed by each algorithm.

perform well. The tree matching algorithm shows good results on the ground truth data. However, the same method on vines whose structure is recovered with other methods provides very poor results. This suggests that the tree matching algorithm is extremely dependent on the structure recovered. This was expected, since the tree matching algorithm uses only the 2D structures of the vines, without passing through a 3D reconstruction and evaluation of the proposed matches. Even with a detailed knowledge of the 2D structure of the vines, the tree matching method is not free from errors, as the reconstructed branch in the top right corner of Figure 6.2b shows. It can be seen that some correct correspondences are not found, as some branches are not covered by back-projected reconstructions.

Figure 6.2c shows the application of RANSAC to the vine in Figure 6.2a. RANSAC finds many branches; however many errors can be observed and some correct correspondences are missed, because some canes are not covered by reconstructions. The most likely explanation is to be sought in the high similarity of the distribution of the variables across the two groups of correspondences. This increases the ambiguity of the possible matches and affects RANSAC's performances. Figure 6.2c also suggests why the test on the connectedness of the structure does not improve the correspondences selected. Given the high number of matchings retained by RANSAC, it is more probable to observe small distances between two canes.

In Figures 6.2d and 6.2e, the reconstructions obtained with maximum likelihood with 200000 and 5000 iterations respectively, are shown. The maximum likelihood performs better than RANSAC and looks similar to the tree matching algorithm in this image. The structure recovered looks complete as well, except for only few branches. However, even maximum likelihood is not free from gross errors. The white arrows in Figure 6.2d and 6.2e point at clearly visible mistakes. The one in the top right corner of Figures 6.2d and 6.2e is probably the same error that was found with the tree matching algorithm. The arrow in the centre of Figure 6.2d points at a reconstruction coming from a wrong correspondence. If Figure 6.2d is compared to Figure 6.2a, it can be seen that there is no branch, in the actual vine, in the position indicated by the arrow in the centre of Figure 6.2d. Therefore, that is a wrong correspondence. There is not much visible difference between the

maximum likelihood with different iterations. The algorithm with 200000 iterations produces slightly better vine models, as I will show in the next section, but it is by far the slowest method.

Figure 6.2f shows the application of the Bayes classifier algorithm. Even if the reconstructed structure is not complete, it is much cleaner, meaning that there are no recognizable wrong correspondences. This is a common feature to the methods using Gibbs sampling: if there is not enough evidence towards the correctness of a correspondence, it is not selected. This results in a less complete structure, but, on the other hand, a more correct one.

Figure 6.2g shows the reconstruction provided by the algorithm averaging the Bayes classifier, Support Vector Machine and the Gradient Boosting. It is very similar to the Bayes classifier method and only after a careful look a difference is found. The arrow in Figure 6.2g points at a cane that is found with the averaging method, but not with the Bayes classifier. This suggests that, in this case, averaging the methods does not provide significantly better results. It is still possible that there are differences between the methods averaged. However they do not reach enough consensus in the averaged method.

The last two methods are the ones performing better among those using Gibbs sampling and the ones having some of the best performances overall. In Figure 6.2h the method using the support vector machine classifier is shown, whereas Figure 6.2i shows the application of the algorithm using gradient boosting. It can be seen that they both find more correct correspondences than the Bayes classifier and the averaged method and no wrong correspondences are visible. Moreover, between the two, the gradient boosting seems to perform better than the support vector machine. It still seems that some correct correspondences are not found, especially if figures 6.2h and 6.2i are compared to the tree matching or the maximum likelihood algorithms. However, the methods using Gibbs sampling are much less dependent on the structure extraction step, i.e. the recovery of the 2D structure of the vines. This means that even on data not coming from the ground truth dataset, their performances are more reliable. Moreover, the figures here shown represent the correspondences found using only a pair of frames. In the actual machine, more images of the same vine are taken, as the robot moves along

the vines. This provides that correspondences and therefore reconstructions are incrementally added to the ones previously computed. In this situation the algorithms using Gibbs sampling show better performances if compared to the remaining methods, as the next section explains.

Analysing the precision - recall values for the methods, it can be seen that given the same α in the methods using Gibbs sampling, gradient boosting has the highest recall. Conversely, the precision is lower. However, given the equality of α, at a 95% level test on the differences, the difference in the recall values is statistically significant ($p-$values ≈ 0.01), whereas the difference in the precision values is never statistically significant ($p-$values ≈ 0.31), if compared to the Bayes classifier and support vector machine methods. There is no statistical difference between the Bayes classifier and support vector machine for $\alpha = 0.9$; in the other cases the recall for support vector machine is statistically higher than that of the Bayes classifier.

6.3 Results on large datasets

In this section I present the results of the application of the methods proposed, to large datasets of real vines. In total there are seven datasets. One is small, showing only four plants and it was used in the initial stages of the project. The other six datasets are bigger. They show images of entire rows of vines that were taken in different vineyards. The vines represented are also of different varieties. There are Sauvignon Blanc, Riesling and Müller Thurgau plants. In Table 6.2 the number of frames and of vine plants for each dataset are reported.

With these datasets, it is not feasible, because it is too time consuming, to check the correctness of correspondences by comparing them to ground truth data. Therefore, some other measures have to be considered to compare the algorithms. To this purpose, the following quantities are defined.

1. The proportion of back-projected pixels that overlap with the actual vine in the image. This quantity is called “Intersection over Union”.

2. The proportion of pixels in the images that are not explained by a 3D

dataset	# frames	# vines
eng-vines	271	4
Sauv. Blanc 1	699	9
Sauv. Blanc 2	699	17
Müller 1	699	10
Müller 2	699	24
Riesling 1	418	15
Riesling 2	699	37
Total	4184	116

Table 6.2: Number of frames for each single camera and number of vine plants for each of the available dataset.

model. This quantity takes into account the missed correct correspondences. In the following it is called "2D Proportion Unexplained".

3. The proportion of back-projected pixels that fall outside the vines. This measure takes into account errors due to reconstructed branches that actually come from wrong correspondences. It is called "3D Proportion Unexplained".

4. The proportion of correspondences selected by the methods over the number of admissible candidate matchings. This quantity on one hand gives informations about the ambiguity solved by the methods; on the other it is proportional to the sum of true and false positives found by each method. However, taking into account the previous error, some qualitative conclusions can still be drawn about the correctness of the chosen correspondences.

The back-projections are considered only on the images coming from the Top camera, as well as the 2D vine structure used to evaluate the correspondences. Because it is difficult to separate vine plants, the evaluation of the previous quantities is made on a per-frame basis and then averaged. However, this procedure may overestimate the 2D and 3D Proportion Unexplained error. Indeed, even if an algorithm finds all the correct correspondences on a vine plant, but in a previous frame finds only few of them, this last error is still

method	IoU	2DProp	3DProp	VCS
BC	0.439745	0.4466525	0.32431	0.07123673
GBM	0.4525125	0.4221025	0.3342725	0.1359093
SVM	0.4647225	0.4135475	0.316095	0.1098979
AVE	0.47793	0.3972925	0.3125675	0.08539775
ML	0.4358554	0.4157696	0.362475	0.2968893
ML_{5000}	0.3313929	0.4999797	0.4393762	0.09926939
RANSAC	0.3201275	0.420665	0.5838225	0.5335714
GM	0.3512232	0.5650393	0.32431	0.04448857

Table 6.3: Summary of the mean values of the observed measures and errors for each method. "IoU" = Intersection over Union, "2DProp" = 2D Proportion Unexplained, "3DProp" = 3D Proportion Unexplained and "VCS" = Vine Correspondences Selected.

included in the quantities considered, although with progressively smaller weights. A more precise evaluation would be obtained if the largest set of correspondences found on a single plant was compared to the actual vine. From the definitions it can be seen that higher values of the Intersection over Union measure, together with lower values for 2D and 3D proportion unexplained, indicate better performances of the methods.

To compare the methods using Gibbs sampling to the others, the following procedure has been adopted. Since the performances of the methods using Gibbs sampling depend on the parameter α in (4.10), for each dataset there are 8 groups of statistics recorded (one group for each value of $\alpha \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$). So, the average on α's of the performances for each dataset has been considered as the overall performance of each method on that dataset. Table 6.3 shows the average values, for each method, of the measures and errors defined.

Figure 6.3 shows the distributions of the Intersection over Union measure. Overall, the methods using Gibbs sampling have better Intersection over Union measure, with higher values than the other algorithms. It seems that averaging the Bayes classifier, gradient boosting and support vector machine contributed to having slightly better 3D reconstructions. The maximum likelihood method with 200000 iterations has performances slightly

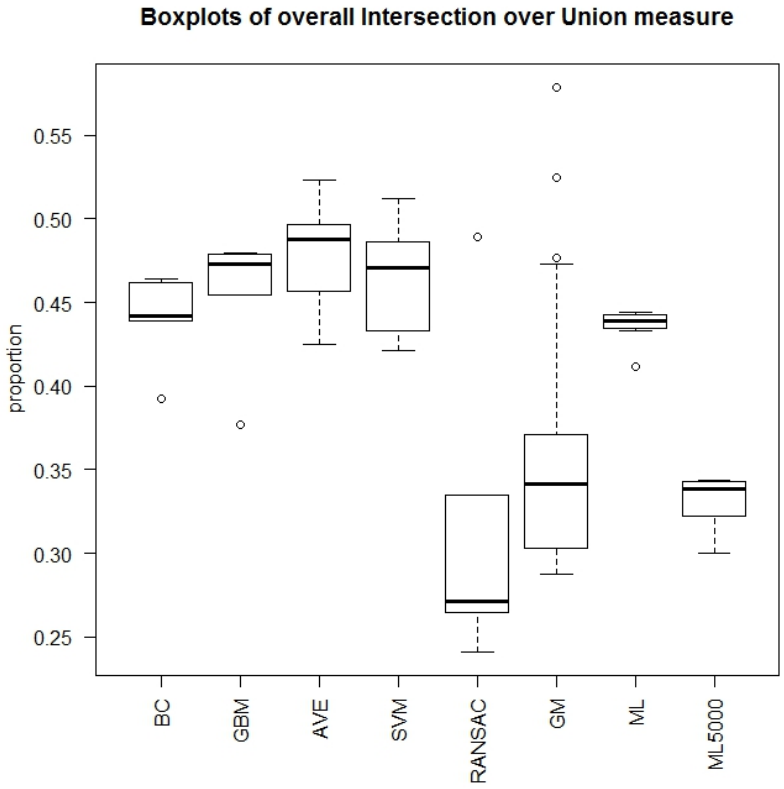


Figure 6.3: Boxplots showing the distributions of the Intersection over Union measure for each method. The dots are considered outliers for the distribution: they are values smaller than $Q_1 - 1.5IQR$ or bigger than $Q_3 + 1.5IQR$, Q_1 and Q_3 being the first and third quartiles for the distribution and $IQR = Q_3 - Q_1$. The thick black line is the median.

lower than Bayes classifier, but when the number of iterations is reduced, the performances decay as well. RANSAC and tree matching have low performances. This is, in case of RANSAC, because it considers a large number of correspondences as correct, even when they are not; in case of tree matching, because it finds fewer correspondences, in general.

Figure 6.4 shows the distributions of the 2D Proportion Unexplained error. The method performing the worst is tree matching. Indeed, it is the algorithm the finds the lowest number of correct correspondences, therefore it has the highest percentage of canes found in the 2D images that do not have a counterpart in the computed reconstruction. RANSAC does better than

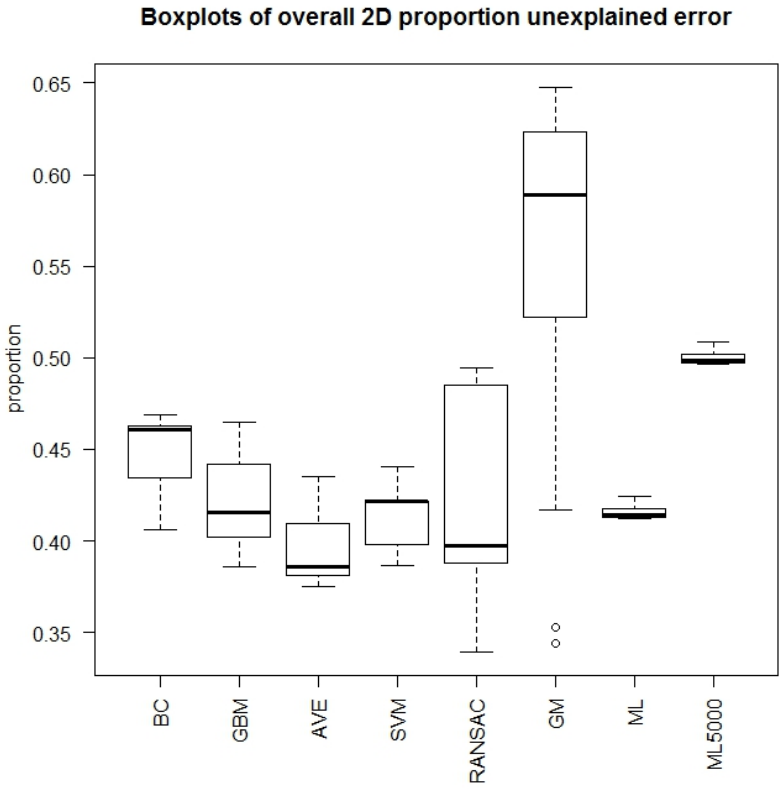


Figure 6.4: Boxplots showing the distributions of the 2D Proportion Unexplained error.

tree matching, but it still has a rather high error. Again, the methods using Gibbs sampling have comparable errors, with the averaged method being slightly better. The averaged method, together with gradient boosting and support vector machine have a lower median, if compared to Bayes classifier. The maximum likelihood with 200000 iterations also has a 2D Proportion Unexplained error comparable to the algorithms using Gibbs sampling and its lower variation shows it has a more constant behaviour.

Figure 6.5 shows the 3D Proportion Unexplained error across the algorithms. RANSAC has the highest error, in this case, and the reason for it is that it considers many incorrect correspondences as correct, as already exemplified, for example, by Figure 6.2c in the previous section. RANSAC has good performances when vines do not have too many canes or they are

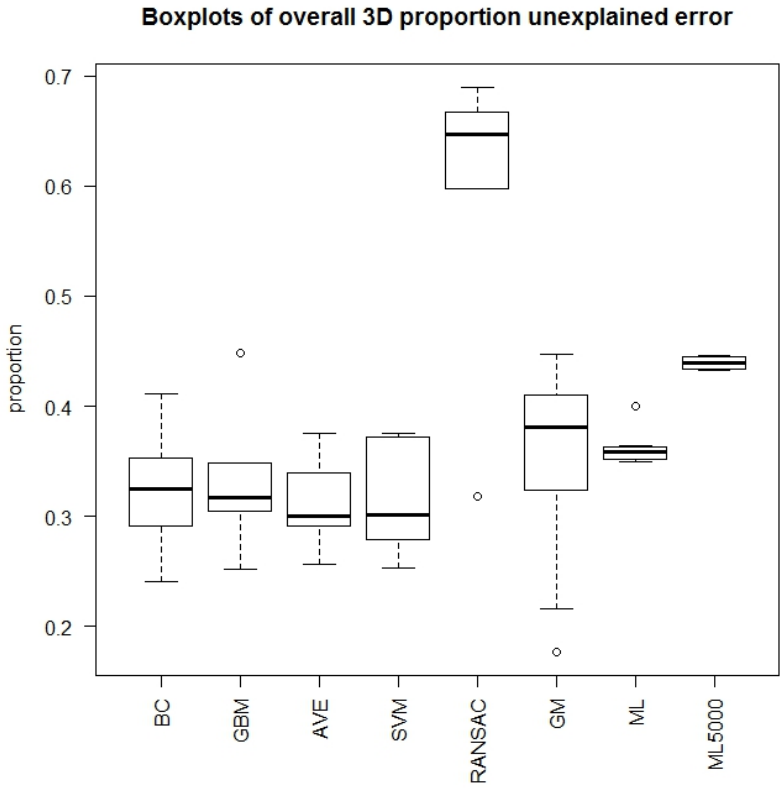


Figure 6.5: Boxplots showing the distributions of the 3D Proportion Unexplained error.

well separated, as shown by Figure 2.18. With more complex vines, there are many more admissible candidate matchings with a higher degree of ambiguity. These may be an explanation of why RANSAC does not perform as well as the methods using Gibbs sampling or maximum likelihood. As a consequence, some of the low values observed for RANSAC in case of the 2D Proportion Unexplained error, may come from the fact that many of the back-projected wrong correspondences cover 2D vines. There is not much difference among the methods using Gibbs sampling and, again, maximum likelihood and tree matching follow. However, gradient boosting and support vector machine have a lower median than Bayes classifier. Overall, if the methods using Gibbs sampling are compared using Intersection over Union, 2D Proportion Unexplained and 3D Proportion Unexplained, there is not

much statistical difference. This is probably due to the overestimation of the errors discussed above. It can also explain why the boxplots for the methods using Gibbs sampling appear similar, but the reconstructions provided by gradient boosting and support vector machine are more complete and accurate and the means and medians are different. If they are compared for fixed α's some significant differences are observable. For example, in terms of the quantities analysed so far, considering the differences in the values, gradient boosting has better performances than Bayes classifier for $\alpha = 0.2, 0.3, 0.4$ ($p-$values $\approx 10^{-4}, 10^{-4}, 0.001$ respectively) and better than support vector machine for $\alpha = 0.2, 0.3$ ($p-$values $\approx 0.001, 0.002$ respectively). In all the other cases it seems that the methods are equivalent. The method using support vector machine is better than Bayes classifier for $\alpha = 0.2, 0.3$ ($p-$values ≈ 0.002).

Finally, in Figure 6.6, the distributions of the Vine Correspondences Selected are shown. RANSAC is the method that finds the highest number of correspondences. However, if the 3D Proportion Unexplained error is taken into account, it must be deduced that the performance shown by RANSAC is biased by the number of false positives. Gradient boosting and support vector machine have the highest number of selected correspondences among the algorithms using Gibbs sampling. Moreover, if those methods are compared at fixed α's, gradient boosting has the highest number of selected correspondences and the difference is always statistically significant. Support vector machine constantly finds more correspondences than Bayes classifier. It is probable that the difference between gradient boosting and support vector machine, if compared to Bayes classifier, is mostly due to the number of true positives. Indeed, if it was caused by false positives, then higher 3D Proportion Unexplained errors would have been observed for gradient boosting and support vector machine, if compared to Bayes classifier. Referring back to (4.10), it can be understood that when α increases, the methods using Gibbs sampling find progressively more matches. Moreover, the algorithms using Gibbs sampling enjoy another property if compared to the others. They are less affected by the complexity of the scene, in the sense that they may find fewer correspondences, but these are mostly correct. On the contrary, maximum likelihood finds more correspondences, but a bigger proportion of those

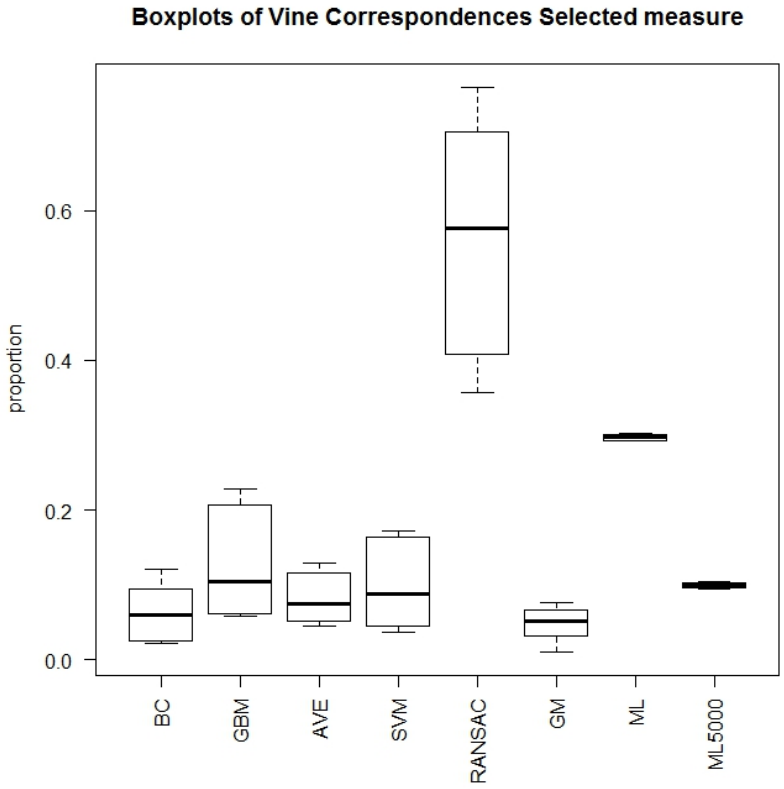


Figure 6.6: Boxplots showing the distributions of the Vine Correspondences selected measure.

is wrong. Figure 6.7 gives an example of the described difference on how the complexity of a scene affects the two classes of methods. Therefore, part of the high value shown by maximum likelihood in the proportion of Vine Correspondences Selected is due to false positives.

Table 6.4 reports the time in seconds spent by each method in finding correspondences between 20, 40 and 100 vine branches. It can be observed that maximum likelihood with 200000 iterations is very slow. Therefore, even if it found mostly correct correspondences, it is not suitable for real-time applications. Even for 20 canes, this version of maximum likelihood takes more than one minute to complete the task. The time taken by Bayes classifier and tree matching are similar and they are the fastest algorithms. Gradient boosting has similar times to support vector machine and they

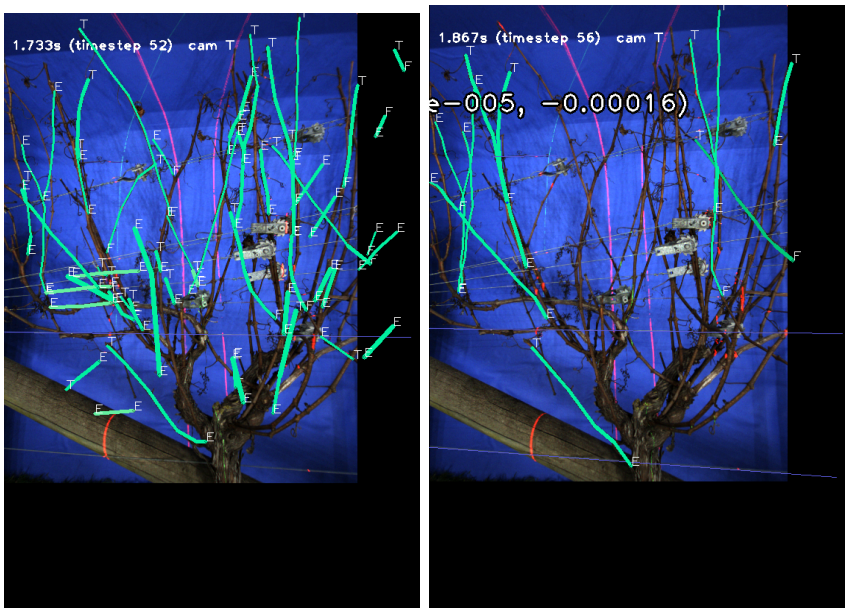


Figure 6.7: Comparison, on the first round of correspondences found, of how the complexity of scenes affects maximum likelihood (left) and gradient boosting (right). Maximum likelihood finds more correspondences, but many of them are wrong. Gradient boosting, instead, finds fewer correspondences, but they seem correct.

have slightly slower times than Bayes classifier. The averaged method was clearly expected to have higher running times if compared to any of the methods being averaged and finally, RANSAC seems to be slightly slower than the averaged algorithm. The times here reported constitute about 1% of the total running time of the automated machine for Bayes classifier, tree matching, maximum likelihood with 5000 iterations and the methods using gradient boosting and support vector machine. For RANSAC and the averaged method it is about 2% and finally, for maximum likelihood with 200000 iterations it is around 64% of the total running time. For all the algorithms, the computational complexity depends on the number of iterations of each method and the number of admissible candidates, therefore giving a complexity of $\mathcal{O}(nk)$, where n is the number of admissible candidates and k is the number of iterations used in each algorithm. Ultimately, the number of admissible candidates, n, depends on the branches seen in each images and

method	20 canes	40 canes	100 canes
BC/ML_{5000}	0.844	6.586	41.581
GBM	0.684	7.15	52.137
SVM	0.807	6.53	57.405
AVE	1.158	9.702	70.577
ML	85.561	428.842	2347.669
RANSAC	1.256	19.135	89.671
GM	1.507	6.781	43.1558

Table 6.4: Running times of the algorithms. This table reports the time spent for finding correspondences with 20, 40 and 100 canes. All the times are in seconds.

therefore on the quantities $|L|, |R|$ and $|T|$. This, in turn, is related to the complexity of the structure. The different operations and steps computed in each method contribute to the differences in their running times.

6.4 Conclusions

In this chapter I have compared the methods that I designed with some adapted state-of-the-art algorithms. The model-based approach appeared to be a sensible choice as an attempt to solve the correspondence problem with vine images. In particular, the algorithms based on gradient boosting and support vector machine brought an improvement on the number of correct correspondences selected and recall rate, together with lower values for the errors considered, therefore providing a more complete and accurate reconstruction of the vine structures if compared to the other methods. Furthermore, they represent an improvement on the initial state of the project, with gradient boosting showing better performances than support vector machine. The quality of the reconstructions provided by the algorithms using Gibbs sampling is the highest, especially in terms of the errors considered. Those algorithms show better performances, in this particular addressed problem, than some commonly used methods to solve correspondence problems, such as RANSAC, maximum likelihood or graph matching. In case of graph matching and RANSAC, the explanation of their lower performances could be found in the self-similarity of the vines and the ambiguity

of possible matchings. In case of RANSAC, proposing improvements presents some difficulties. Its main problem is a high acceptance rate. This cannot be solved by restricting the acceptance criterion, because, given the similarity of the distribution of the variables considered, it will imply rejecting inliers with a high probability. It could be possible that including in RANSAC a more complex model linking the variables, could bring improvements. However, estimating this model in the hypothesis generation step could slow RANSAC considerably. In case of the tree matching algorithm, it was observed that it had better performances on the set of ground truth data. This could imply that if the structure extraction step in the project provided results similar to ground truth, tree matching could be a method that may be considered to solve the problem addressed in this thesis. For maximum likelihood, the problem is again a higher acceptance rate as correct of correspondences that, in reality, are not.

Looking at the reconstructions provided by maximum likelihood with 200000 iterations and the method using gradient boosting, it can be deduced that the right proportion of vine correspondences selected must be strictly less than 30% and greater than 20% of the total number of admissible candidates. Therefore, there is still a small percentage of correct correspondences that are not found by the gradient boosting method. This may be due to the fact that the variables defined to model the branches cannot solve this residual amount of ambiguity. As seen in Chapter 3, the variables have similar distributions across the two groups of correspondences. This fact affects the performances of any classifier. Identifying new features, for example from a more precise 2D structure recovery of the vines, or incorporating depth sensors in the pruning machine could help overcoming this issue in the future.

With this chapter the objectives 1., 4. and 5. of the list in Section 1.4 have been fully achieved. Original methods were developed, analysed and compared against suitably modified state-of-the-art methods. Code written in C++ was produced, tested and integrated in the pruning machine. From the results presented, the gradient boosting method can be considered the best one for the problem I investigated.

Chapter VII

Conclusions

7.1 Conclusion

This thesis has described proposed model-based approaches aimed at solving correspondence problems in the case of complex branched structures. The motivation at the basis of the research undertaken was the designing of an automated pruning machine for grapevines. The best methods have provided a sensible improvement on prior research, in terms of vines reconstructed, with only a slight increase in running time. In case of the method using gradient boosting, the improvement is about 90% on prior research (Botterill *et al.* [20]). They also proved more effective if compared to state-of-the-art methods suitably modified to be applied to the considered problem.

To design algorithms that could achieve a higher number of correct correspondences, a first step in the research was to define and analyse suitable variables of branches (Chapter 3). Two main classes of quantities describing geometric and topological attributes of branches were therefore considered. Of these two classes, the geometrical variables resulted more discriminative in deciding about the correctness of a correspondence. On the contrary, the topological variables were not generally as useful, but this is probably due to a bias introduced by the 3D reconstructing algorithm. The definition and analysis of those variables constituted a first novelty brought by my research. Especially the extension of the notion of shape to 3D polylines.

The acquired knowledge about the branches was useful to propose algorithms that look for correct correspondences using only information about the vines and the physical characteristics of the machine. In Chapter 4, which constitutes the second novel contribution of my research, I discussed a first group of proposed methods. They estimate marginal probabilities of a

match being correct and then select the correspondences. This class of methods includes: Bayes classifier, support vector machine, gradient boosting and an algorithm averaging those.

In Chapter 5, another group of proposed algorithms used state-of-the-art approaches with suitable modifications in order to be applied to my specific problem. Specifically, RANSAC, a maximum likelihood algorithm and a graph matching approach were proposed.

The methods proposed were compared on precision - recall performances on a set of ground truth data and on bigger datasets, with suitably defined errors.

Even though the methods I proposed are specific to a unique problem and application, I believe that they can be useful in other applications as well. One example of an application is correspondence problems with other branched structures, for example blood vessels, as in [153], or pulmonary airway trees, as in [50]. Moreover, the extension on the notion of shape for 3D curves can be further developed and applied in many other fields.

7.2 Future work

Better reconstructions would be achieved by integrating new aspects or variables in the algorithms that classify a match as correct or incorrect. An example is represented by the laser lines that exist in the machine but were never integrated in the correspondence step. The lasers give information about the depth of the branches and this could be used to further limit the ambiguity of admissible candidate matchings. This may be the right direction to follow, as the variables considered were not able to discriminate between the two groups of correspondences in a complete way.

Another improvement that needs to be considered is a better 2D structure recovery from the images. As the structure of the vines was the input of my problem, a better recovered set of branches, together with the relations among them, implies more precise and less ambiguous matches.

A reinforcement learning system could also be considered as a future direction. It can be developed in two different ways. On one hand the pruning machine in its current state can be used and then followed by an expert

human that rates the performance of the machine. Based on this, the machine can learn from the errors it made. On the other hand, the ambiguous branches that are not matched can be assigned to a different class, rather than correct or incorrect matches. Then, a new classifier could be built on these data and integrated in the system.

In the meanwhile, while the Automated Pruning Project is still approaching its final development, the gradient boosted method can be reliably adopted as the reference matching algorithm for this application.

References

[1] R. J. Adler, J. E. Taylor, (2005), Random Fields and Geometry, Springer-Verlag, Berlin.

[2] R. J. Adler, (2008), *Some new random field tools for spatial analysis*, Stochastic Environmental Research and Risk Assessment, Vol. 22, No. 6.

[3] R. Anderson, B. Stenger and R. Cipolla, (2011), *Color photometric stereo for multicolored surfaces*, Proc. ICCV 2011.

[4] E. Arias-Castro, D. L. Donoho, X. Huo, (2006), *Adaptive Multiscale Detection of Filamentary Structures in a Background of Uniform Random Points*, The Annals of Statistics, Vol. 34, No. 1, 326 - 349.

[5] C. Arora, A. Globerson, (2013), *Higher Order Matching for Consistent Multiple Target Tracking*, Proceedings ICCV 2013.

[6] E. Artin, (1955), Elements of Algebraic Geometry, course notes for the course in Algebraic Geometry held at New York University.

[7] E. Artin, (1957), Geometric Algebra, Interscience Publishers, Inc., New York.

[8] A. Baak, M. Mueller, G. Bharaj, H.-P. Seidel, C. Theobalt, (2013), *A Data-Driven Approach, for Real-Time Full Body Pose Reconstruction from a Depth Camera*, Proc. ICCV 2013.

[9] A. O. Bălan, (2003), *Voxel Carving and Colouring - Constructing a 3D Model of an Object from 2D Images*, downloadable from the author's website.

[10] T. F. Banchoff, (1967), *critical points and curvature for embedded polyhedra*, Journal of Differential geometry, Vol. 1, No. 3 - 4, 245 - 256.

[11] R. G. Baraniuk, M. B. Wakin, (2009), *Random Projections on Smooth Manifolds*, Foundations of Computational Mathematics, Vol. 9, No. 1.

[12] E. Begelfor, M. Werman, (2005), *How to put Probabilities on Homographies*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 10, 1666 - 1670.

[13] P. N. Belhumeur, (1993), *A Binocular Stereo Algorithm for Reconstructing Sloping, Creased and Broken Surfaces in the Presence of Half-occlusion*, Proceedings IEEE Fourth International Conference on Computer Vision, 431 - 438.

[14] P. Belhumeur, (1996), *A bayesian Approach to Binocular Stereopsis*, International Journal of Computer Vision, Vol. 13, No. 3, 237 - 260.

[15] P. Belhumeur, A. Yuille, (1995), *Recovering Object Surfaces from Viewed Changes in Surface Texture Patterns*, Proceedings of IEEE fifth International Conference on Computer Vision, 876 - 881.

[16] P. N. Belhumeur, D. Mumford, (1992), *A Bayesian Treatment of the Stereo Correspondence Problem Using Half-Occluded Regions*, in Computer Vision and Pattern Recognition. Proceedings CVPR '92, 1992 IEEE Computer Society Conference.

[17] A. G. Belyaev, (2008), *A Note on Invariant Three-Point Curvature Approximations*, Departmental Bulletin Paper.

[18] S. Birchfield, C. Tomasi, (1996), *Depth Discontinuities by Pixel-by-Pixel Stereo*, Stanford University Technical Report STAN-CS-TR-96-1573, July 1996.

[19] T. Botterill, R. Green, S. Mills, (2012), *Reconstructing partially visible models using stereo vision, structured light and the g2o framework*, in Proceedings of IVCNZ 2012.

[20] T. Botterill, R. Green, S. Mills, (2014), *A decision-theoretic formulation for sparse stereo correspondence problems*, Proceedings 3D Vision.

[21] D. Bradley, and T. Beeler, (2013), *Local Signal Equalization for Correspondence Matching*, Proc. ICCV 2013.

[22] L. Breiman, (1996), *Bagging Predictors*, Machine Learning, 123 - 140.

[23] R.G. Brown, J.G. Chase, C.E. Hann, (2012), *A pointwise smooth surface stereo reconstruction algorithm without correspondence*, Image and Vision Computing, accepted manuscript.

[24] P. Buhlmann, B. Yu, (2003), *Boosting with the L_2 Loss: Regression and Classification*, Journal of the American Statistical Association, Vol. 28, 324 - 339.

[25] F. Cao, J.L. Lisani, J.M. Morel, P. Musé, F. Sur, (2008), A Theory of Shape Identification, Springer-Verlag, Berlin Heidelberg.

[26] C. Wang, L. Wang and L. Liu, (2013), *Improving Graph Matching via Density Maximization*, Proc. ICCV 2013.

[27] M. Cho, J. Lee, K. M. Lee, (2010), *Reweighted random walks for graph matching*, ECCV 2010.

[28] M. Cho, K. M. Lee, (2012), *Progressive graph matching: Making a move of graphs via probabilistic voting*, CVPR 2012.

[29] M. Cho, K. Alahari, J. Ponce, (2013), *Learning Graphs to Match*, Proc. ICCV 2013.

[30] L. Cehovin, M. Kristan and A. Leonardis, (2011), *An adaptive coupled-layer visual model for robust visual tracking*, Proc. ICCV 2011.

[31] O. Chum, J. Matas, (2005), *Matching with PROSAC - Progressive Sample Consensus*, Computer Vision and Pattern Recognition, Vol. 1, 220 - 226.

[32] R. Cipolla, A. Blake, (1992), *Surface Shape from the Deformation of Apparent Contours*, International Journal of Computer Vision, Vol. 9, No. 2, 83 - 112.

[33] D. Cohen-Steiner, J.-M. Morvan, (2003), *Restricted Delaunay Triangulations and Normal Cycle*, Proceedings of the nineteenth annual symposium on Computational Geometry.

[34] D. Cohen-Steiner, H. Edelsbrunner, (2005), *Inequalities for the Curvature of Curves and Surfaces*, Proceedings of the twenty-first annual symposium on Computational Geometry.

[35] D. Cohen-Steiner, H. Edelsbrunner, D. Morozov, (2006), *Vines and Vineyards by Updating Persistence in Linear Time*, Proceedings of the twenty-second annual symposium on Computational Geometry, 119 - 126.

[36] J. B. Conway, (1985), A course in functional analysis, Springer-Verlag, New York.

[37] P. Debevec, C. J. Taylor, J. Malik, (1996), *modeling and Rendering Architecture from Photographs: A hybrid geometry-and image-based approach*, SIGGRAPH 96 conference proceedings.

[38] D. Delaere, C. Smets, P. Suetens, G. Marchal, (1991) *Knowledge-based system for the three dimensional reconstruction of the blood vessels from two angiographic projections*, Journal of medical and biological engineering and computing, Vol. 29, No. 6.

[39] D. Dey, L. Mummert, R. Sukthankar, (2012), *Classification of plant structures from uncalibrated image sequences*, IEEE Workshop on the Applications of Computer Vision.

[40] L. Di Stefano, M. Marchionni, S. Mattoccia, (2004), *A fast area-based stereo matching algorithm*, Image and Vision Computing, Vol. 22.

[41] M. Dongbo, L. Jiangbo and M. N. Do, (2011), *A revisit to cost aggregation in stereo matching: How far can we reduce its computational redundancy?*, Proc. ICCV 2011.

[42] O. Duchenne, A. Joulin, J. Ponce, (2013), *A graph-matching Kernel for Object Categorization*, Proceedings ICCV 2013.

[43] E. Dunn, B. Clipp and J. M. Frahm, (2011), *A geometric solver for calibrated stereo egomotion*, Proc. ICCV 2011.

[44] S.Y. Elhabian, H. Rara and A. A. Farag, (2011), *Towards accurate and efficient representation of image irradiance of convex-Lambertian objects under unknown near lighting*, Proc. ICCV 2011.

[45] M. C. Er, (1987), *A Fast Algorithm for Generating Set Partitions*, The Computer Journal, Vol. 31, No. 3, 283 - 284.

[46] Y. Eshet, S. Korman, E. Ofek, S. Aviden, (2013), *DSCH - Matching Patches in RGBD Images*, Proceedings ICCV 2013.

[47] B. Ettinger, T. Passerini, S. Perotto, L.M. Sangalli, (2012), *Regression models for data distributed over non-planar domains*, MOX Report No. 22/2012.

[48] J. Fayad, C. Russell and L. Agapito, (2011), *Automated articulated structure and 3D shape recovery from point correspondences*, Proc. ICCV 2011.

[49] J. Feng, W. Yichen, T. Litian, Z. Chao and S. Jian, (2011), *Salient object detection by composition*, Proc. ICCV 2011.

[50] A. Feragen, P. Lo, M. de Bruijne, M. Nielsen, F. Lauze, (2013), *Toward a Theory of Statistical Tree-Shaped Analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 35, No. 8, 2008 - 2021.

[51] M. A. Fischler, R. C. Bolles, (1981), *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, Communications of the ACM, Vol. 24, No. 6.

[52] P. T. Fletcher, C. Lu, S. M. Pizer, S. Joshi, (2004), *Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape*, IEEE Transactions on Medical Imaging, Vol. 23, No. 8.

[53] V. Fragoso, P. Sen, S. Rodriguez and M. Turk, (2013), *EVSAC: Accelerating Hypotheses Generation by Modeling Matching Scores with Extreme Value Theory*, Proc. ICCV 2013.

[54] Y. Freund, Y. Mansour, R. E. Schapire, (2004), *Generalization bounds for averaged classifiers*, The Annals of Statistics, Vol. 32, No. 4, 1698 - 1722.

[55] J. H. Friedman, (2001), *Greedy function approximation: a gradient boosting machine*, The Annals of Statistics, 29, 1189 - 1232.

[56] J. H. Friedman, B. E. Popescu, (2008), *Predictive Learning via Rule Ensembles*, The Annals of Statistics, Vol 2, No. 3, 916 - 954.

[57] A. Geiger, M. Roser, R. Urtasun, (2010), *Efficient Large-Scale Stereo Matching*, Asian Conference on Computer Vision.

[58] D. Geiger, B. Ladendorf, A. Yuille, (1995), *Occlusions and Binocular Stereo*, International Journal of Computer Vision, Vol. 14, 211 - 226.

[59] U. Grenander, (1963), Probabilities on Algebraic Structures, Almqvist & Wiksells, Uppsala.

[60] U. Grenander, (1981), Abstract Inference, John Wiley and Sons, Inc., New York.

[61] U. Grenander, (1989), *Advances in Pattern Theory*, The Annals of Statistics, Vol. 17, No. 1.

[62] U. Grenander, (1993), General Pattern Theory. A Mathematical Study of Regular Structures, Clarendon Press, Oxford.

[63] L. C. Grove, (2002), Classical Groups and Geometric Algebra, Americal Mathematical Society, Providence, Rhode Island.

[64] G. Lin, P. H. Cong, A. Robles-Kelly and Z. Jun, (2011), *Material-specific user colour profiles from imaging spectroscopy data*, Proc. ICCV 2011.

[65] Y. Hacohen, E. Shechtman and D. Lischinski, (2013), *Deblurring by Example Using Dense Correspondence*, Proc. ICCV 2013.

[66] S. Hadfield and R. Bowden, (2011), *Kinecting the dots: Particle based scene flow from depth sensors*, Proc. ICCV 2011.

[67] R. Hartley, A. Zisserman, (2004), Multiple View Geometry in Computer Vision, Cambridge University Press, Cambridge.

[68] T. Hastie, R. Tibshirani, J. Friedman, (2009), The Elements of Statistical Learning, Springer-Verlag, New York.

[69] S. Hawe, M. Kleinsteuber and K. Diepold, (2011), *Dense disparity maps from sparse disparity measurements*, Proc. ICCV 2011.

[70] P. Heise, S. Klose, B. Jensen, A. Knoll, (2013), *PM-Huber: PatchMatch with Huber Regularization for stereo matching*, Proceedings ICCV2013.

[71] Y. Hel-Or, H. Hel-Or, E. David, (2013), *Fast Template Matching in Non-linear Tone-Mapped Images*, Proceedings ICCV 2013.

[72] W. V. D. Hodge, D. Pedoe, (1953), Methods of Algebraic Geometry, Cambridge University Press, Cambridge.

[73] J. Huang, D. Mumford, (1999), *The Statistics of Natural Images and Models*, Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 541 - 547.

[74] H. Ishikawa, D. Geiger, (2006), *Rethinking the Prior Model for Stereo*, in Proceedings of the 9th European Conference on Computer Vision, 526 - 537.

[75] J.-H. Kim, Y. Dai, L. Hongdong, D. Xin and K. Jonghyuk, (2013), *Multi-view 3D Reconstruction from Uncalibrated Radially-Symmetric Cameras*, Proc. ICCV 2013.

[76] J. B. Jessen, (2009), *Real time sparse and dense stereo in an early cognitive vision system using CUDA*, Master's thesis, University of Southern Denmark.

[77] R. A. Johnson, D. W. Wichern, (1982), Applied Multivariate Statistical Analysis, Pearson Prentice Hall, Upper Saddle River, New Jersey.

[78] I. M. Johnstone, (2001), *On the distribution of the largest eigenvalue in principal component analysis*, The Annals of Statistics, Vol. 29, No. 2, 295 - 327.

[79] S. C. Joshi, M. I. Miller, U. Grenander, (1997), *On the Geometry and Shape of Brain Sub-Manifolds*, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 11, 1317 - 1343.

[80] S. Joshi, S. Pizer, P. T. Fletcher, P. Yushkevic, A. Thall, J. S. Marron, (2002), *Multiscale Deformable Model Segmentation and Statistical Shape Analysis Using Medial Descriptions*, IEEE Transactions on Medical Imaging, Vol. 21, No. 5.

[81] Kaggle, `https://www.kaggle.com/c/higgs-boson`, accessed 4 June 2015.

[82] Kaggle, `https://www.kaggle.com/c/malware-classification/forums/t/13897/first-place-code-and-documents`, accessed 5 June 2015.

[83] K. Kanatani, (1990), Group-Theoretical Methods in Image Understanding, Springer-Verlag, Berlin-Heidelberg.

[84] K. Kanatani, (1998), *Statistical optimization and geometric inference in computer vision*, Philosophical Transactions of the Royal Society A, Vol. 356, No. 1740, 1303 - 1320.

[85] M. Kang, P. Cournède, J. Quadrat, P. de Reffye, (2006), *A Stochastic Language for Plant Topology*, in Proceedings of the 2006 International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications.

[86] D. Kendall, (1984), *Shape Manifolds, Procrustean Metrics and Complex Projective Spaces*, Bulletin of the London Mathematical Society, Vol. 16.

[87] P. Kohli, (2005), *Efficiently solving dynamic Markov random fields using graph cuts*, Tenth IEEE International Conference on Computer Vision, 2005, ICCV 2005.

[88] V. Kolmogorov, R. Zabih, (2005), *Graph Cut Algorithms for Binocular Stereo with Occlusions*, in Mathematical Models In Computer Vision: The Handbook, Springer, 423 - 438.

[89] P. Kontchieder, S. Rota Bulò, H. Bischof, M. Pelillo, (2011), *Structured Class-Labels in Random Forests for semantic image labelling*, Proceedings ICCV 2011.

[90] S. Korman, S. Avidan, (2011), *Coherency Sensitive Hashing*, Proceedings ICCV 2011.

[91] R. Kumar, J. O. Talton, S. Ahmad, T. Roughgarden, S. R. Klemmer, (2011), *Flexible Tree Matching*, Proceedings of IJCAI.

[92] C. Lau, W. Heidrich and R. Mantiuk, (2011), *Cluster-based color space optimizations*, Proc. ICCV 2011.

[93] M. Leordeanu, M. Hebert, (2005), *A spectral technique for correspondence problems using pairwise constraints*, Proceedings ICCV 2005.

[94] M. Leordeanu, A. Zanfir, C. Sminchisescu, (2013), *Locally Affine Sparse-to-Dense Matching for Motion and Occlusion Estimation*, Proceedings ICCV 2013.

[95] Leutenegger, S. and Chli, M. and Siegwart, R.Y., (2013), *BRISK: Binary Robust invariant scalable keypoints*, Proc. ICCV 2013.

[96] C. Li, O. Deussen, Y.-Z. Song, P. Willis, P. Hall, (2011), *Modelling and Generating Moving Trees from Video*, ACM Trans. Graph. 30, 6, Article 127.

[97] P. Li, Q. Wang, L. Zhang, (2013), *A Novel Earth Mover's Distance Methodology for Image Matching with Gaussian Mixture Models*, Proceedings ICCV 2013.

[98] L. Huei-Hung, L. Yuping and G. Medioni, (2013), *Aerial 3D reconstruction with line-constrained dynamic programming*, Proc. ICCV 2013.

[99] E. Lobaton, R. Vasudevan, R. Alterovitz and R. Bajcsy, (2013), *Robust topological features for deformation invariant image matching*, Proc. ICCV 2013.

[100] S. Mairhofer, S. Zappala, S. R. Tracy, C. Sturrock, M Bennett, (2012): *RooTrak: Automated Recovery of Three Dimensional Plant Root Architecture in Soil from X-Ray Microcomputed Tomography Images Using Visual Tracking*, Plant Physiology, Vol. 158, pp. 561-569.

[101] R. Manduchi, C. Tomasi, *Ambiguity in Stereo Matching.*

[102] D. Marr, T. Poggio, (1976), *Cooperative Computation of Stereo Disparity*, Science, Vol. 194, No. 4262, 283 - 287.

[103] J. Marroquin, S. Mitter, T. Poggio, (1987), *Probabilistic Solution of Ill-Posed Problems in Computational Vision*, Journal of the American Statistical Association, Vol. 82, No. 397, 76 - 89.

[104] S. Maybank, (1993), Theory of Reconstruction from Image Motion, Springer - Verlag, Berlin - Heidelberg.

[105] S. Maybank, P. Beardsley, (1994), *Applications of Invariants to Model-Based Vision*, Journal of Applied Statistics, Vol. 21, No. 5, 431 - 457.

[106] B. McCane, (2014),*Case deletion for fundamental matrix computation*, Proceedings IVCNZ 2014.

[107] M. Menor, G. Poisson, and K. Baek, (2012), *Probabilistic prediction of protein phosphorylation sites using kernel machines*, Proceedings of the 27th ACM Symposium on Applied Computing Conference Track on Bioinformatics and Computational Systems Biology, pp.1393-1398.

[108] M. I. Miller, L. Younes, (1999): *Group actions, homeomorphisms, and matching: a general framework*, Proceedings of the IEEE Workshop on Statistical and Computational Theories of Vision.

[109] D. Mumford, A. Desolneux, (2010), Pattern Theory. The Stochastic Analysis of Real-World Signals, A K Peters, Natick, Massachusetts.

[110] D. Mumford, B. Gidas, (2001), *Stochastic Models for Generic Images*, Quarterly of Applied Mathematics, Vol. 59, No. 1, 85 - 111.

[111] R. Narasimha, (2006), *Markov Random Field for Stereo*, tutorial.

[112] J. Neira, J. D. Tardós, (2001), *Data Association in Stochastic Mapping Using the Joint Compatibility Test*, IEEE Transactions on Robotics and Automation, Vol. 17, No. 6.

[113] Z. Ni, T. F. Burks, (2013), *Plant or Tree Reconstruction Based on Stereo Vision*, ASABE Annual International Meeting.

[114] S. Nowozin, C. Rother, S. Bagon, B. Yao, Toby Sharp, P. Kohli, (2013), *Decision Tree Fields*, Proceedings ICCV 2013.

[115] A. Ogale, Y. Aloimonos, (2005), *Shape and the Stereo Correspondence Problem*, International Journal of Computer Vision, Vol. 63, No. 3.

[116] K. Omasa, F. Hosoi, A. Konishi, (2007), *3D LiDAR imaging for detecting and understanding plant responses and canopy structure*, Journal of Experimental Botany, Vol 58, No. 4.

[117] P. Oskoui-Fard, H. Stark, (1988), *Tomographic Image Reconstruction Using the Theory of Convex Projections*, IEEE Transactions on Medical Imaging, Vol. 7, No. 1.

[118] A. Paproki, J. Fripp, O. Salvado, X. Sirault, S. Berry, R. Furbank, (2011), *Automated 3D segmentation and analysis of cotton plant*, 2011 International Conference on Digital Image Computing: Techniques and Applications.

[119] J. Park, H. Kim, Y.W. Tai, M.S. Brown, and I. Kweon, (2011) *High quality depth map upsampling for 3D-TOF cameras*, Proc. ICCV 2011.

[120] H. S. Park and Sheikh, Y., (2011), *3D reconstruction of a smooth articulated trajectory from a monocular image sequence*, Proc ICCV 2011.

[121] X. Pennec, (2004), *Probabilities and Statistics on Riemmanian Manifolds: A Geometric approach*, Rapport de recherche.

[122] J. C. Platt, (1999), *Probabilistic Outputs for Support Vector Machine and Comparisons to Regularized Likelihood Methods*, in ADVANCES IN LARGE MARGIN CLASSIFIERS, 61 - 74.

[123] R. Raguram and J.M. Frahm, (2011), *RECON: Scale-adaptive robust estimation via Residual Consensus*, Proc. ICCV 2011.

[124] D. Raviv, A. Dubrovina, R. Kimmel, (2011), *Hierarchical Matching of Non-rigid Shapes*, International Conferences on Scale Space and Variational Methods in Computer Vision (SSVM).

[125] C. P. Robert, G. Casella, (2005), Monte Carlo Statistical Methods, Springer-Verlag New York Inc., Secaucus, NJ, USA.

[126] E. Rodola, A. Torsello, T. Harada, Y. Kuniyoshi and D. Cremers, (2013), *Elastic Net Constraints for Shape Matching*, Proc. ICCV 2013.

[127] C. A. Rothwell, A. Zisserman, D. A. Forsyth, J. L. Mundy, (1995), *Planar Object Recognition using Projective Shape Representation*, International Journal of Computer Vision, Vol. 16, 59 - 99.

[128] M. Rouhani and A. D. Sappa, (2011), *Correspondence free registration through a point-to-model distance minimization*, Proceedings ICCV 2011.

[129] Y. A. Rozanov, (1982), Markov Random Fields, Springer-Verlag, Berlin.

[130] H. Sadeghi, P. Moallem, S. Amirhassn Monadjemi, (2008), *Feature Based Dense Stereo Matching using Dynamic Programming and Color*, International Journal of Information and Mathematical Sciences, Vol. 4, No. 3.

[131] M. Sarkis, K. Diepold, (2008), *Sparse stereo matching using belief propagation*, 15^{th} IEEE International Conference on Image Processing.

[132] S. Satkin, M. Herbert, (2013), *3DNN: Viewpoint Invariant 3D Geometry Matching for Scene Understanding*, Proceedings ICCV 2013.

[133] O. Saurer, K. Koser, J.-Y. Bouguet and M. Pollefeys, (2013), *Rolling Shutter Stereo*, Proc. ICCV 2013.

[134] D. Scharstein, C. Pal, (2007), *Learning Conditional Random Fields for Stereo*, IEEE Conference on Computer Vision and Pattern Recognition.

[135] D. Scharstein, R. Szeliski, (2002), *A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms*, International Journal of Computer Vision, Vol. 43, No. 1 - 3, 7 - 42.

[136] R. Schneider, W. Weil, (2008), Stochastic and Integral Geometry, Springer, Berlin.

[137] J. G. Semple, G. T. Kneebone, (1952), Algebraic Projective Geometry, Clarendon Press, Oxford.

[138] E. Sharon, D. Mumford, (2004): *2D-Shape Analysis using Conformal Mapping*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[139] C. Sumetphong, S. Tangwongsan (2012), *An optimal approach towards recognizing broken Thai characters in OCR systems*, International Conference on Digital Image Computing Techniques and Applications (DICTA).

[140] C. Sun, H. Talbot, S. Ourselin, T. Adriaansen, *Embedded Voxel Colouring*, in Proceedings of the Seventh International Conference on Digital Image Computing: Techniques and Applications, DICTA 2003.

[141] R. Szeliski, (2010), Computer Vision: Algorithms and Applications, Springer, Berlin.

[142] C.-H. Teng, Y.-S. Chen, W.-H. Hsu, (2006), *Constructing a 3D trunk model from two images*, Graphical Models, 69, 33 - 56.

[143] M. E. Tipping, (2001), *Sparse Bayesian Learning and the Relevance Vector Machine*, Journal of Machine Learning, 1, 211 - 244.

[144] P. H. S. Torr, A. Zissermann, (2000), *MLSEAC: A New Robust Estimator with Application to Estimating Image Geometry*, Computer Vision and Image Understanding, Vol. 78, No. 1, 138 - 156.

[145] B. Triggs, P. F. McLauchlan, R. I. Hartley, A. W. Fitzgibbon, (2000), *Bundle Adjustment - A Modern Synthesis*, in Proceedings Vision Algorithms '99.

[146] A. Trouvé, (1995), *An Infinite Dimensional Group Approach for Physics based Models in Pattern Recognition*, International Journal of Computer Vision, Vol. 01.

[147] K. Turner, S. Mukherjee, D. M. Boyer, (2013), *Sufficient Statistics for Shapes and Surfaces*, Annals of statistics.

[148] B. Ummenhofer and T. Brox, (2013), *Point-Based 3D Reconstruction of Thin Objects*, Proc ICCV 2013.

[149] M. Vaillant, M. I. Miller, L. Younes, A. Trouvé, (2004), *Statistics on diffeomorphisms via tangent space representations*, NeuroImage, 23, 161 - 169.

[150] V. N. Vapnik, (1998), Statistical Learning Theory, John Wiley & Sons Inc., Hoboken, NJ, USA.

[151] O. Veksler, (2005), *Stereo correspondence by dynamic programming on a tree*, CVPR 2005.

[152] M. P. Wand, M. C. Jones, (1994) *Multivariate plugin bandwith selection*, Computational Statistics, 9, 97 - 116.

[153] H. Wang, J. S. Marron, (2007), *Object Oriented Data Analysis: Sets of Trees*, The Annals of Statistics, Vol. 35, No. 5.

[154] P. Weinzaepfel, J. Revaud, Z. Harchaoui, C. Schmid, (2013), *DeepFlow: Large displacement optical flow with deep matching*, Proceedings ICCV 2013.

[155] A. Weiss, D. Hirshberg and M.J. Black, (2011), *Home 3D body scans from noisy image and range data*, Proceedings ICCV 2011.

[156] D. H. Wolpert, W. G. Macready, (1997), *No Free Lunch Theorems for Optimization*, IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, 67 - 82.

[157] Z. Wu, R. Leahy, (1993), *An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, 1101 - 1113.

[158] D. Xia, F. Yang, S. Xu, D, Li, Q. Li, (2012), *Constructing three dimensional plant stems model from images*, Optical Engineering, Vol. 51.

[159] J. Yan, Y. Tian, H. Zha, X. Yang, Y. Zhang, S. M. Chu, (2013), *Joint optimization for consistent multiple graph matching*, Proceeding ICCV 2013.

[160] M. Ye, X. Wang, R. Yang, L. Ren and M. Pollefeys, (2011), *Accurate 3D pose estimation from a single depth image*, Proc. ICCV 2011.

[161] J. S. Yedidia, W. T. Freeman, Y. Weiss, (2001), *Understanding Belief Propagation and its Generalization*, International Joint Conference on Artificial Intelligence.

[162] L. Younes, (1999), *Optimal matching between shapes via elastic deformations*, Image and Vision Computing, Vol. 17, 381 - 389.

[163] Q. Zhang, X. Song, X. Shao, H. Zhao, R. Shibasaki, (2013), *Learning Graph Matching: oriented to Category Modeling from Cluttered Scenes*, Proceedings ICCV 2013.

[164] Y. Zhang, M. Brady, S. Smith, (2001), *Segmentation of Brain MR Images Through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm*, IEEE Transactions on Medical Imaging, Vol. 20, No. 1.

[165] W. Zhang, J. Kosecka, (2006), *Generalized RANSAC Framework for Relaxed Correspondence Problems*, Proceedings of the Third International Symposium on 3D Data Processing, Visualization and Transmission.

[166] Y. Zheng, S. Gu, H. Edelsbrunner, C. Tomasi, P. Benfey, (2011), *Detailed reconstruction of 3D plant root shape*, 2011 IEEE International Conference on Computer Vision.

[167] X. Zhou, R. Wang, (2006), *Symmetric Pixel-Group Based Stereo Matching for Occlusion Handling*, Proceedings of the 18th Conference on Pattern Recognition, ICPR06.

[168] S. Zhu, D. Mumford, (2006), *A Stochastic Grammar of Images*, Foundations and Trends in Computer Graphics and Vision.

[169] S. C. Zhu, Y.N. Wu, D.B. Mumford, (1997), *Minimax entropy principle and its applications to texture modeling*, Neural Computation, Vol. 9, No. 8.

[170] J. B. Zin, R. Dupont, A. Bartoli, (2013), *A General Dense Image Matching Framework Combining Direct and Feature-based Costs*, Proceedings ICCV 2013.

[171] Z. Ma, K. He, Y. Wei, J. Sun and E. Wu, (2013), *Constant Time Weighted Median Filtering for Stereo Matching and Beyond*, Proc ICCV 2013.